



## 2010 ITSO Parallel Sysplex and High Availability Update

[www.ibm.com/redbooks](http://www.ibm.com/redbooks)

Frank Kyne, ITSO Poughkeepsie



© 2010 IBM Corporation. All rights reserved.



## Welcome

My background.....

My funny accent - please remind me when I start going too fast!

Questions?? Please ask as I go along.

**PLEASE** complete the evaluation forms:

- Especially, if you are not happy, please say WHY.
- If you feel additional clarification is needed on any topics, please indicate that as well.
- Please indicate which bits you liked, which you didn't like, and if there was something missed that you would have liked to hear...

Thanks to everyone that helped me with this material, especially Bert DeBeer from Netherlands

## Topics

**Sysplex enhancements in z/OS 1.12**

**z196 and sysplex**

**Latest Coupling Facility enhancements**

**CF Service Level management**

**Implementing SMF use of Log Streams**

**Latest Mean Time to Recovery enhancements**

**z/OS Resiliency tools:**

- Predictive Failure Analysis
- Runtime Diagnostics

**Miscellaneous**

## Timeline

<b>09:??</b>	<b>Start</b>
<b>10:30</b>	<b>Coffee</b>
<b>12:15</b>	<b>Lunch</b>
<b>15:00</b>	<b>Afternoon Coffee</b>
<b>17:00</b>	<b>Finish</b>

## Trademarks

The following terms are trademarks of the International Business Machines Corporation in the United States, other countries, or both:

IBM has two registered trademarks for the branding of ITSO publications. These registered marks are for the text word "IBM Redbooks" and the Redbooks logo. In a nutshell, the term Redbooks must always be used in the plural form (for both text and logo) since IBM only owns the registered mark for the plural form. Usage must follow the guidelines below:

### Using the term Redbooks in written text

Redbooks are only to be referred to in the plural form, NEVER in the singular.

For the initial reference (first occurrence), you must use "IBM Redbooks®" and include "IBM" as well as the ®. For instances thereafter you may use "Redbooks" without "IBM" preceding the word or ® following it.

### Correct usage for written text :

In this IBM Redbooks® publication we will explore.....(® symbol required for 1st usage)  
This Redbooks publication will show you.....(2nd usage or later - no ® or "IBM" needed)

### Using the logo:

Redbooks (logo)



### OTHER ITSO PUBLICATIONS - Marks not yet registered

Trademark registration is a lengthy process and until we are officially registered, we cannot use the ® symbol. For those terms/logos in process, we will be using the ™ symbol. In contrast to the ® symbol (placed in the lower right hand corner), the ™ symbol is placed in the upper right hand corner. Please see examples below:

Redpaper ™  
Redpapers ™  
Redwiki ™  
Redwikis ™



The following terms are trademarks of other companies:

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

Other company, product, or service names may be trademarks or service marks of others.



©2010 IBM Corporation. All rights reserved.

## Trademarks

The following terms are trademarks of the International Business Machines Corporation in the United States, other countries, or both:

eServer™	DS8000™	RACF®
pSeries®	FICON®	REXX™
z/OS®	GDPS®	RMF™
z/VM®	HyperSwap™	System z™
z/VSE™	IBM®	SystemPac®
z9™	IMST™	Tivoli®
BatchPipes®	MVS™	VTAM®
CICS®	Parallel Sysplex®	WebSphere®
DB2®	PAL™	
DFSMSHsm™	PR/SM™	
DFSMSrmm™	Redbooks®	

The following terms are trademarks of other companies:

Intel, Intel Inside (logos), MMX, and Pentium are trademarks of Intel Corporation in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

SET, SET Secure Electronic Transaction, and the SET Logo are trademarks owned by SET Secure Electronic Transaction LLC.

CA is a trademark of Computer Associates

Other company, product, and service names may be trademarks or service marks of others.



©2010 IBM Corporation. All rights reserved.



International Technical Support Organization

[ibm.com/redbooks](http://ibm.com/redbooks)

## Sysplex enhancements in z/OS 1.12



© 2010 IBM Corporation. All rights reserved.

[ibm.com/redbooks](http://ibm.com/redbooks)

International Technical Support Organization



## z/OS 1.12 Sysplex enhancements

- Enhancements to XCF exploiter monitoring with additional automatic actions for "Critical" XCF members
- New capabilities in Sysplex Failure Management for hung structure processes
- Enhancements to the REALLOCATE command
- Sublist notification delay enhancement
- Additional XCF FUNCTIONS parameters
- New sysplex-related health checks
- System Logger enhancements
- Enhancements to CFSizer



©2010 IBM Corporation. All rights reserved.



## XCF enhancements for hung/stalled address spaces



## XCF enhancements

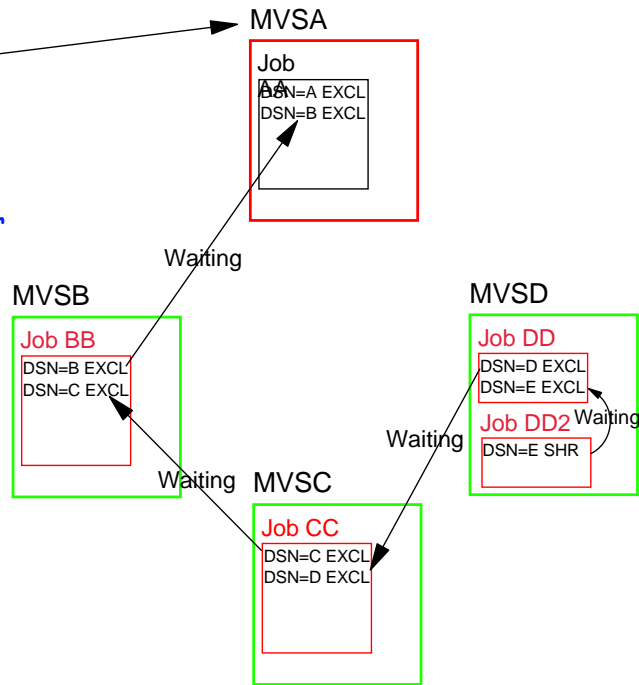
The design mode of operation for a sysplex is that all resources should be accessible from all members of the sysplex and any work should be able to run anywhere in the sysplex.

This means that it should be possible to maintain application availability even if one or more of the systems or subsystems in the plex are down. And the impact of an outage should be smaller, because you only lose 1/x of the users of each application in case of one failure.

The downside is that if a system dies:

- It is probably holding resources that will be needed by another member of the sysplex.
- It cannot release those resources because it is dead.

The longer a dead system remains in the sysplex (holding resources), the larger is the impact on other systems.



## Addressing sympathy sickness situations

To address this situation, there have been a steady stream of enhancements to XCF and XES over recent z/OS releases:

MEMSTALLTIME	z/OS 1.8
SSUMLIMIT	z/OS 1.9
SFM and AutoIPL	z/OS 1.10
<b>SFM and BCPii</b>	<b>z/OS 1.11</b>
Changed default SSUM action	z/OS 1.11
FDI consistency with SPINRCVY	z/OS 1.11
Critical member support	z/OS 1.12
CFSTRHANGTIME	z/OS 1.12

This is FANTASTIC, you have to use this

## Identifying stalls

XCF in each system in a sysplex writes a "heartbeat" to the Sysplex Couple Data Set every 2-3 seconds.

- Other systems in the sysplex use this to identify a system that is possibly not functioning and then take action against that system. This capability is SIGNIFICANTLY enhanced by XCF use of BCPII in z/OS 1.11.

## Identifying stalls

Stalled member detection identifies a stalled address space and issues an operator message. This is triggered when an address space takes too long to retrieve a message that XCF has told it is ready for collection.

**MEMSTALLTIME**, introduced in z/OS 1.8, takes that a step further by optionally terminating a stalled XCF member that is causing sympathy sickness.

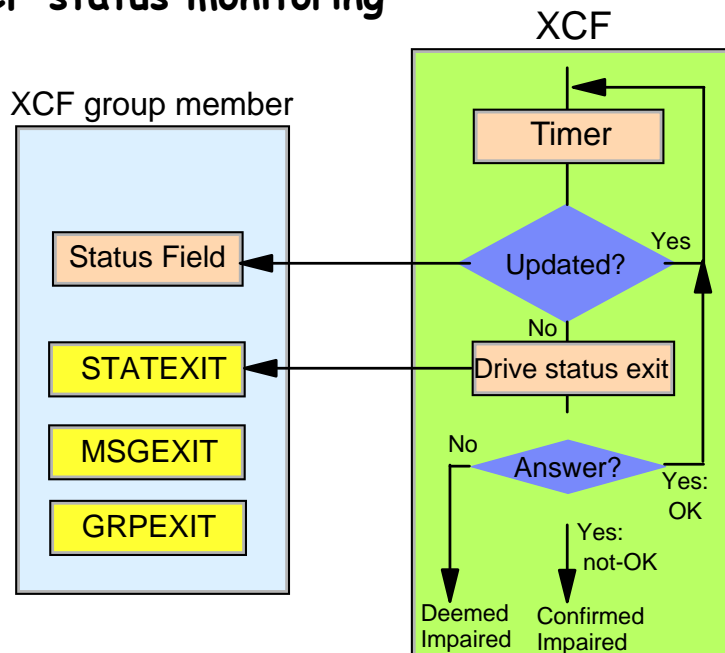
However, if an address space is not sending or receiving messages, there was nothing to help XCF spot that the address space has a problem.

## z/OS 1.12 enhancements

**z/OS 1.12 adds two new capabilities in this area:**

- XCF will now monitor the (optional) status field of an XCF member, as well as the progress of its message, group, and status exits to detect hung or damaged address spaces. If it detects such an address space, a console message will be issued, identifying the impaired address space.
- IN ADDITION, XCF exploiters have the option to tell XCF what action it should take if they get hung or damaged. This is called Critical member support.

## The XCF status monitoring





## Detecting stalled XCF members

XCF exploiters have 3 exits that they can use in relation to XCF:

- Message exit. A small program that is scheduled to retrieve newly-arrived messages for that member from XCF.
- Group exit. This is used by XCF to inform members about the activity of other members in the same XCF group. The use of this exit is optional.
- Status exit. A program, provided by the XCF member, that XCF can schedule to check on the status of the member and inform XCF of the results. This is also optional.

All 3 exits existed prior to z/OS 1.12, but the monitoring of the member's status field and progress of the Group and Status exits is new in 1.12.

## Status monitoring

Pre-R12 D

XCF,G,xxx,ALL output:

Message exits

Group exit

```

INFO FOR GROUP SYSGRS MEMBER SC04 ON SYSTEM SC04

MENTOKEN: 040002A4 00030004      ASID: 0007      SYSID: 04001F80
INFO: CURRENT      COLLECTED: 09/07/2010 17:37:13.363043

SIGNALING SERVICE
MSGO ACCEPTED: 180328343 NOBUFFER: 0
MSGO XFER CNT: 104979019 LCL CNT: 74743840 BUFF LEN: 956
MSGO XFER CNT: 1 LCL CNT: 0 BUFF LEN: 4028
MSGO XFER CNT: 610 LCL CNT: 44 BUFF LEN: 36796
MSGO XFER CNT: 275589 LCL CNT: 19197 BUFF LEN: 40892
MSGO XFER CNT: 291404 LCL CNT: 19430 BUFF LEN: 44988

SENDPND  RESPPND  COMPLTD  MOSAVED  MISAVED
MESSAGE TABLE: 0 0 0 0 0

MSGI RECEIVED: 181114769 PENDINGQ: 0
MSGI XFER CNT: 106335308 XFERTIME: 714

IO BUFFERS      DREF  PAGEABLE
MSGI PENDINGQ: 0 0 0
SYMPATHY SICK: 0

EXIT 055C5700: 09/07/2010 17:37:05.176315 NA 00:00:00.000126
EXIT 055C5500: 09/07/2010 17:37:04.083820 NA 00:00:00.000235
EXIT 01D6BF00: 09/07/2010 17:37:04.337116 NA 00:00:00.000169
EXIT 055C4500: 09/07/2010 17:37:05.176302 NA 00:00:00.000137
EXIT 05880A80: 09/07/2010 17:37:05.176311 NA 00:00:00.000129
EXIT 0587F880: 09/07/2010 17:37:04.084029 OM 00:00:00.000009
EXIT 0256F900: 09/07/2010 17:37:04.084034 OM 00:00:00.000010
EXIT 0256F700: 09/07/2010 17:37:04.337158 NA 00:00:00.000127

GROUP SERVICE
EVNT RECEIVED: 155 PENDINGQ: 0

EXIT 0285D0A8: 08/27/2010 21:32:12.576542 02 00:00:00.000002
  
```

## Status monitoring

### Enhancements in z/OS 1.12:

- XCF monitors the group, message, and status exits for a stall condition. If any of these exits don't complete within a certain time, XCF deems the member to be impaired.
- Additionally, a member (through its status exit) can declare itself to be impaired.
- For a member that either declares itself to be ill, or XCF determines that it is ill from monitoring the exits, XCF now informs the operator about the "non-operational" state of a member (in addition to informing the other group members). However, XCF will NOT take any action against those members (unless they are known to be causing sympathy sickness).
- For critical members (that is, those that specify CRITICAL=YES on the IXCJOIN macro), if the impaired condition persists, XCF WILL take action against the member at the level specified on the TERMLEVEL keyword.

## Status monitoring

R12 D XCF,G,xxx,ALL  
output:

### New information

- Provided for ALL members, regardless of which exits they define or whether they specify CRITICAL=YES

INFO FOR GROUP SYSGRS MEMBER #@\$A ON SYSTEM #@\$A

FUNCTION: GLOBAL ENQ PROCESSING  
MEMTOKEN: 0A00008D 00020004 ASID: 0007 SYSID: 0A000280  
INFO: CURRENT COLLECTED: 09/07/2010 17:22:01.202971

ATTRIBUTES JOINED: 08/13/2010 13:12:27.068243

JOIN TASK ASSOCIATION  
CRITICAL MEMBER  
LOCAL CLEANUP NOT NEEDED  
TERMLEVEL IS SYSTEM  
MEMSTALL RESOLUTION IS SYSTEM TERMINATION AFTER 165 SECONDS

EXITS DEFINED: MESSAGE, GROUP, STATUS

#### SIGNALING SERVICE

MSGO ACCEPTED:	1203304	NOBUFFER:	0
MSGO XFER CNT:	659730	LCL CNT:	116845
MSGO XFER CNT:	49	LCL CNT:	36
MSGO XFER CNT:	31049	LCL CNT:	0
MSGO XFER CNT:	211928	LCL CNT:	0
MSGO XFER CNT:	183667	LCL CNT:	0

# Status monitoring

R12 D XCF,G,xxx,ALL

output:

```

SENDPND  RESPPND  COMPLTD  MOSAVED  MISAVED
MESSAGE TABLE:  0      0      0      0      0
CRITICAL:        0      0      0      0      0

MSGI RECEIVED:  1220487  PENDINGQ:    0
MSGI XFER CNT:  1107809  XFERTIME:    169

          IO BUFFERS          DREF  PAGEABLE  CRITICAL
MSGI PENDINGQ:  0              0      0          0
SYMPATHY SICK:  0

EXIT 04751D00: 09/07/2010 17:22:00.071341 OM 00:00:00.000004
EXIT 0210AB00: -                NA -
EXIT 0474FF00: 09/07/2010 17:22:00.075016 MC 00:00:00.000005
EXIT 04751100: 09/07/2010 17:22:00.075011 NA 00:00:00.000012

GROUP SERVICE
EVNT RECEIVED:  165  PENDINGQ:    0

EXIT 0252E190: 09/04/2010 13:43:08.845127 02 00:00:00.000002

MONITOR SERVICE
STAT INTERVAL:  10  STATUS: NORMAL
STAT DETECTED:  08/13/2010 13:12:27.072708
LAST VERIFIED:  09/07/2010 17:21:52.033152
EXIT 0253B0F0: 09/07/2010 17:21:52.033137 SM 00:00:00.000003
    
```

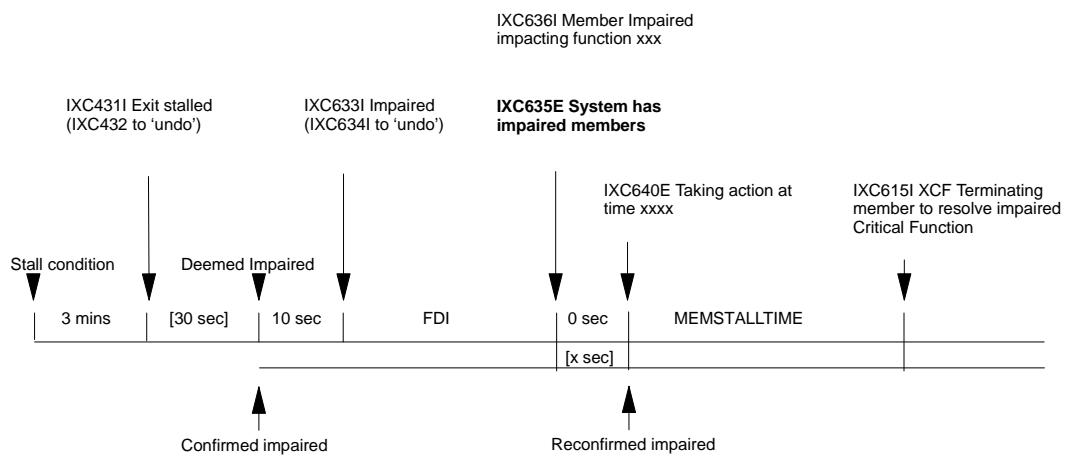
- Message exits

- Group exit

- Status exit

**And yet MORE new information**

# Status monitoring



Stall condition = At least 1 exit stalled for 30 seconds or work item on head of queue for 30 seconds

Deemed impaired = IXC431I issued and for last 30 seconds, either all scheduled user exits stalled or no user exits scheduled

## Status monitoring

### Possible new member status':

#### - [CONFIRMED STATUS MISSING](#)

- The member's status exit indicates that the member is not operating normally and is considered to be in a status update missing condition.

#### - [CONFIRMED IMPAIRED](#)

- A member is "confirmed impaired" if the member is in a Confirmed Status Missing condition long enough to impact the normal operation of the member.

#### - [DETECTED STATUS MISSING](#)

- The member's status exit did not execute in a timely fashion, and XCF deems the member to be in a status update missing condition.

#### - [DEEMED IMPAIRED](#)

- A member is "deemed impaired" if all of its exits processing user-related requests appear to be stalled and impacting the normal operation of the member function.

## "Critical" members

**IN ADDITION to the new monitoring and notification function, in z/OS 1.12, XCF has support for "Critical" members (as determined by the product itself):**

#### - [Specify CRITICAL=YES on IXCJOIN when connecting to an XCF group](#)

- The developer has determined that the function is critical to the group and/or the system, and should be terminated (for the good of the group or the system or the sysplex) when impaired.
- This would normally be used together with the member's status field and status exit, to help XCF identify stall or hang conditions in a timely way.

### Termination level:

- [Product developer specifies \(using the TERMLEVEL keyword on the IXCJOIN macro\) what termination action to take \(task, jobstep, address space or even system\)](#)

## Processing for a stalled critical member

When a stall condition is detected:

- XCF will:

• Present messages

Only for CRITICAL=YES

- Give the member some time to recover (depending on FDI)
- Redrive the exits
- Allow some more time (depending on MEMSTALLTIME)
- Issue an abend OOC reason 020F000D to create a LOGREC entry and then schedule a dump to gather diagnostic information
- Execute the action specified on the TERMLEVEL

- The timing and intervals between the above actions depends on:

- Internal XCF time settings
- Failure detection interval (INTERVAL on COUPLE stmt in COUPLExx), or the default (4\*spintime +5 seconds).
- MEMSTALLTIME setting in SFM policy

## Commands, new and changed messages

There are no changes to system commands (however the output from existing command is enhanced with additional information)

Some new messages:

- IXC633I "member is impaired"
- IXC634I "member is no longer impaired"
- IXC635E "system has impaired members"
- IXC636I "impaired member impacting function"

All of these have extensive descriptive text in the Messages Manual - strongly recommended to review this.

Ensure these messages are added to your automation product.

## Commands, new and changed messages

### Some changed messages:

- IXC431I "member stalled" (includes status exit)
- IXC640E "going to take action"
- IXC615I "terminating to relieve impairment"
- IXC333I "display member details"
- IXC101I, IXC105I, IXC220W "system partitioned"

## Sample invocation

Sample where  
**CRITICAL=YES**  
 member's status  
 exit sets status to  
**IMPAIRED**

- Note that message says that "SFM will take action" even if an SFM policy is not active

```
IXC633I GROUP GROUP1 MEMBER MEM1 JOB XEBHBZA7 ASID 002A 246
CONFIRMED IMPAIRED AT 09/08/2010 19:12:25.617516 ID: 1.3
  LAST MSGX:                                0 STALLED          0 PENDINGQ
  LAST GRPX: 09/08/2010 19:14:03.179947      0 STALLED          0 PENDINGQ
  LAST STAX: 09/08/2010 19:16:07.932102      0 STALLED
IXC636I GROUP GROUP1 MEMBER MEM1 JOB XEBHBZA7 ASID 002A 247
  IMPAIRED, IMPACTING CRITICAL FUNCTION TESTCASE XCJHBZA7
IXC640E IMPAIRED XCF GROUP MEMBERS ON SYSTEM SY1 IMPACTING SYSPLEX 248
SFM WILL TAKE ACTION AT 09/08/2010 19:24:19
IEA045I AN SVC DUMP HAS STARTED AT TIME=19.23.51 DATE=09/08/2010 249
FOR ASIDS(0006,0001,002A)
ERROR ID = SEQ00020 CPU81 ASID0001 TIME19.23.51.5
QUIESCE = YES
IEA794I SVC DUMP HAS CAPTURED: 250
DUMPID=001 REQUESTED BY JOB (*MASTER*)
DUMP TITLE=COMPON=XCF,COMPID=5752SCXCF,ISSUER=IXCM2REC,MODULE=I
          XCS1DCM,ABEND=S000C,REASON=020F000D
IXC615I GROUP GROUP1 MEMBER MEM1 JOB XEBHBZA7 ASID 002A 253
SFM TERMINATING JOB STEP TASK TO RELIEVE IMPAIRMENT CONDITION
IXC640E IMPAIRED XCF GROUP MEMBERS ON SYSTEM SY1 IMPACTING SYSPLEX 257
SFM IS TAKING ACTION
IXC634I GROUP GROUP1 MEMBER MEM1 JOB XEBHBZA7 ASID 002A 260
NO LONGER IMPAIRED.
TERMINATING AT 09/08/2010 19:24:31.035682 ID: 1
```

## GRS use of the enhancement

GRS is currently the only exploiter of this function.

GRS Star sicknesses that can be detected by this function:

- Signaling, GRS does not respond to XCF signals
- Inability to schedule an SRB in a timely manner
- Inability of GLOBAL and Qscan to process requests

GRS ring Sickness:

- Signaling, GRS does not respond to XCF signals

Based on the GRS TERMLEVEL selection, XCF will waitstate the system if GRS meets the impaired critical system member criteria.

## Migration

Note that the action specified on the TERMLEVEL keyword will be carried out EVEN IF SFM IS NOT ACTIVE:

- If SFM is not active, the greater of the FDI or 2 minutes will be used as the MEMSTALLTIME value in the processing for this function.
- The closest you can get to turning this new function (actioning impaired CRITICAL=YES members) OFF is to specify a very large MEMSTALLTIME value. HOWEVER, that will apply to ALL stalled member processing.
- Recommended MEMSTALLTIME value:
  - If you have automation or procedures that will take action based on the XCF messages, then set MEMSTALLTIME to 2-5 minutes.
  - If you will NOT take any action by automation on the messages, then set it to a very low value and let SFM handle it.



## Coexistence

Update your automation policies for the new and changed messages

Toleration APAR OA31619 should be applied on z/OS V1R10 and z/OS V1R11 systems to prevent incorrect and misleading messages as a result of down-level systems misunderstanding the new sysplex partitioning reason codes.

## Summary

The new status monitoring function may potentially work with any address space:

- But to get the benefit, you must have a process to take some action based on the messages that XCF puts out.

In addition, the ability for address spaces to tell XCF what action to take if they become impaired should:

- Reduce occurrences of sympathy sickness
- Improve diagnosis
- Improve mean time to recovery (MTTR)





## XES Structure Hang Detection



## Structure hang detection

Just as the new XCF member status monitoring support is intended to address sympathy sickness situations, equally, the new CFSTRHANGTIME support is intended to address situations where one party in an XES process is stopping the process from progressing because it is not responding.

## XES structure hang detection (pre z/OS 1.12)

Coordinated and timely processing by all participants is required for:

- Structure rebuild / duplexing
- Connector failure / disconnect recovery
- User synch points

Processing hang is detected and reported after 2 minutes by XES messages (IXLO40E and IXLO41E).

However, what happens if there is no correct reaction to the messages?

- Sympathy sickness: hang of other users of the structure
- Potentially resulting in an application or system outage

## New capabilities in Sysplex Failure Management

The existing XES hang detection function for structure-related events is unchanged; what is new is SFM support that exploits the information surfaced by XES.

SFM is enhanced to take the following recovery actions in an attempt to resolve the hang (depending on the type of problem):

- Stopping the rebuild
- Stopping signaling path (XCF signaling Structures only)
- Forcing a disconnect ( XCF signaling Structures only)
- Terminating the connector task
- Terminating the connector address space
- Partitioning the connector system

## Activation and usage

There is a new SFM policy keyword that lets you control this function - **CFSTRHANGTIME**.

- Specifies the number of seconds before SFM takes action (after the 2 minutes XES was waiting), or NO when no automatic action should be taken (default).

To exploit this new capability, you must start an SFM policy and specify some value other than NO for **CFSTRHANGTIME**.

## An example

D XCF, COUPLE

IXC357I 11.38.53 DISPLAY XCF 628

SYSTEM #@\$3 DATA

INTERVAL	OPNOTIFY	MAXMSG	CLEANUP	RETRY	CLASLEN
165	168	2000	15	10	956

SSUM ACTION	SSUM INTERVAL	SSUM LIMIT	WEIGHT	MEMSTALLTIME	
ISOLATE		0	720	50	2

CFSTRHANGTIME  
5

DEFAULT USER INTERVAL: 165  
 DERIVED SPIN INTERVAL: 165  
 PARMLIB USER OPNOTIFY: + 3

MAX SUPPORTED CFLEVEL: 17

```

DATA TYPE(SFM) REPORT(YES)

DEFINE POLICY NAME(SFM02) REPLACE(YES)
CONNFAIL(YES)
SYSTEM NAME(*)
  ISOLATETIME(0)
  SSUMLIMIT(720)
SYSTEM NAME(#@$1)
  WEIGHT(80)
SYSTEM NAME(#@$2)
  WEIGHT(90)
SYSTEM NAME(#@$3)
  WEIGHT(50)
  CFSTRHANGTIME(5)
  
```

## Changed and New messages

### Changed messages

- IXC220W new waitstate codes
- IXC307I reason for stopping a path
- IXC357I CFSTRHANGTIME added
- IXC360I status added
- IXC467I reason for stopping a path
- IXC522I reason for stopping rebuild
- IXC614I CFSTRHANGTIME after policy activation

### New messages

- IXL047I response no longer expected
- IXL048I response no longer expected
- IXL049E hang resolution action
- IXL050I connector has not responded in time

## Migration

### Review automation policy

- For changed messages
- For replaced messages
  - IXL042I and IXL043 replaced by IXL047I and IXL048I regardless of whether you specify CFSTRHANGTIME or even whether SFM is active or not.

**Update SFM policy with CFSTRHANGTIME keyword and specify a value between 0 and 1800 (seconds)**

## Coexistence

Toleration APAR OA30880 for z/OS V1R10 and z/OS V1R11 makes reporting of the **CFSTRHANGTIME** keyword with **IXCMIAPU** utility possible on those releases:

- However the **SFM** function is not rolled back to previous releases
- Issuing a **D XCF,C** on a back-level system will not show any **CFSTRHANGTIME** value. Remember that the output from **D XCF,C** relates to the system that the command was issued on. If you specify a **CFSTRHANGTIME** value for a pre-R12 system, it will be ignored.

## CRITICAL=YES for structure connectors

XCF exploiters normally connect to an XCF group directly, using an **IXCJOIN** macro.

However, any CF structure that contains lock entries (lock structures and serialized list structures) also needs to be able to use an XCF group so information related to lock contention for that structure can be routed between the systems.

In this case, **XES** connects to an XCF group called **IXCLOxxx** automatically on behalf of the connector.

For example:

## CRITICAL=YES for structure connectors

Lock structure

```

D XCF,STR,STRNM=ISGLOCK
IXC360I 00.35.22 DISPLAY XCF 591
STRNAME: ISGLOCK
STATUS: ALLOCATED
EVENT MANAGEMENT: MESSAGE-BASED
TYPE: LOCK
POLICY INFORMATION:
  POLICY SIZE   : 33280 K
  ...

ACTIVE STRUCTURE
-----
ALLOCATION TIME: 07/18/2010 16:00:09
CFNAME       : FACIL03
COUPLING FACILITY: 002097.IBM.02.00000001DE50
              PARTITION: 0E  CPCID: 00

ACTUAL SIZE   : 33 M
STORAGE INCREMENT SIZE: 1 M
USAGE INFO    TOTAL    CHANGED    %
LOCKS:       4194304
PHYSICAL VERSION: C64BF2EF 1CB53800
LOGICAL  VERSION: C64BF2EF 1CB53800
SYSTEM-MANAGED PROCESS LEVEL: 8
XCF GRPNAME  : IXCLO007
DISPOSITION  : DELETE
ACCESS TIME  : 0
MAX CONNECTIONS: 32
# CONNECTIONS : 3

```

XCF Group

## CRITICAL=YES for structure connectors

Using GRS as an example, GRS specifies CRITICAL=YES when it joins the XCF group called SYSGRS.

But what about the IXCLO007 group that is used for lock contention communication? This is also associated with GRS (because it is used exclusively for the GRS lock structure), but the IXCJOIN is done by XES rather than by GRS...

So how does GRS get the benefits of CRITICAL=YES for possible problems related to the IXCLO007 group?

## CRITICAL=YES for structure connectors

If a structure connector specifies **CRITICAL=YES** and **TERMLEVEL** on the **IXLCONN** macro, this causes **XES** to specify **CRITICAL=YES** and **TERMLEVEL** when it issues the **IXCJOIN** for the **IXCLO007** group.

- XES identifies the structure connector (GRS in this example) as the address space that is associated with that XCF group

## CRITICAL=YES for structure connectors

If you display the XCF group associated with the lock structure, you now see which address space is actually using the services of that structure, and the structure associated with the group:

```
D XCF,G,IXCLO007,ALL
IXC333I 14.31.58 DISPLAY XCF 622
INFORMATION FOR GROUP IXCLO007
MEMBER NAME:      SYSTEM:      JOB ID:      STATUS:
M633              #@$A              GRS          ACTIVE
M657              #@$2              GRS          ACTIVE
M658              #@$3              GRS          ACTIVE

INFO FOR GROUP IXCLO007 MEMBER M633 ON SYSTEM #@$A

FUNCTION: FOR STR ISGLOCK
MEMTOKEN: 0A00008F 00040003      ASID: 0007      SYSID: 0A000280
INFO: CURRENT      COLLECTED: 09/22/2010 14:31:58.931103

ATTRIBUTES      JOINED: 08/13/2010 13:12:25.702714
JOIN TASK ASSOCIATION
CRITICAL MEMBER
LOCAL CLEANUP NOT NEEDED
TERMLEVEL IS SYSTEM
MEMSTALL RESOLUTION IS SYSTEM TERMINATION AFTER 165 SECONDS
EXITS DEFINED: MESSAGE
...
```

## CRITICAL=YES for structure connectors

So now we have two (independent) levels of monitoring in relation to the GRS lock structure:

- Structure Hang Detection, in case a connector is not responding to a structure-related event. AND
- XCF status monitoring for processing related to the XCF signalling for lock contention in that structure.

## Summary

You do not need to do anything to enable this CRITICAL=YES support:

- It is invoked by the connector automatically (so far, GRS is the only exploiter).
- The function is independent of whether you enable CFSTRHANGTIME or not.
- The function is NOT dependent on SFM being active.





International Technical Support Organization

[ibm.com/redbooks](http://ibm.com/redbooks)

**CFRM enhancements**



© 2010 IBM Corporation. All rights reserved.

[ibm.com/redbooks](http://ibm.com/redbooks)

International Technical Support Organization



## **CFRM enhancements**

### **REALLOCATE command**

- Test version of the REALLOCATE command
- Report the results of a previously-executed SETXCF START,REALLOCATE command

### **Sublist notification delay**

- You can now control the delay between notifications



©2010 IBM Corporation. All rights reserved.

## REALLOCATE command

A powerful command, that saves a lot of work, time, and confusion during reconfigurations, and attempts to place structures in the "right" CF as per the PREFLIST in the CFRM policy.

The **START REALLOCATE** command should **ALWAYS** be used (together with the **MAINTMODE** command) for CF management tasks (emptying and re-populating):

- Much better (smarter) than POPCF
- MUCH better than SETXCF START,REBUILD,LOC=NORMAL

## REALLOCATE command

However, the use of **START,REALLOCATE** raises two common questions:

- Can I get a summary of what happened? The output is scattered through the syslog making it difficult to see exactly what happened.
- A test version would nice, so we can see what would happen **BEFORE** we actually do it.

**The answer:**

- **D XCF,REALLOCATE,REPORT**
  - Summarizes the information from the last SETXCF REALLOCATE command
- **D XCF,REALLOCATE,TEST**
  - Tells you what REALLOCATE would do IF you ran it
  - Report function will be called under the covers at the end of execution

## Report example (first part only)

```
D XCF,REALLOCATE,REPORT
IXC347I 13.23.23 DISPLAY XCF 787

THE REALLOCATE PROCESS STARTED ON 09/01/2010 AT 09:40:34.58.
THE REALLOCATE PROCESS ENDED ON 09/01/2010 AT 09:40:42.88.
-----
STRUCTURE(S) WITH AN ERROR/EXCEPTION CONDITION

STRNAME: SYSTEM_OPERLOG                                INDEX: 13
  1 REALLOCATE STEP(S): REBUILD
  EXCEPTION ON SYSTEM #@$3 ON 09/01/2010 AT 09:40:37.18.
  THE REQUIRED REBUILD WAS STOPPED DUE TO
  CONNECTOR SPECIFIC REASON
-----
STRUCTURE(S) WITH A WARNING CONDITION

NONE
-----
STRUCTURE(S) REALLOCATED SUCCESSFULLY

STRNAME: IFASMF_TYPCICS                                INDEX: 369
  1 REALLOCATE STEP(S): REBUILD
  COMPLETED ON SYSTEM #@$A ON 09/01/2010 AT 09:40:41.46.
```

## Report example

```
D XCF,REALLOCATE,TEST
IXC347I 18.46.54 DISPLAY XCF 097

COUPLING FACILITY STRUCTURE ANALYSIS PERFORMED FOR REALLOCATE TEST.
-----
STRUCTURE(S) WITH AN ERROR/EXCEPTION CONDITION

NONE
-----
STRUCTURE(S) WITH A WARNING CONDITION

NONE
-----
```

## Report example

```

STRUCTURE(S) REALLOCATED SUCCESSFULLY

STRNAME: IGWLOCK00                                INDEX: 0
SIMPLEX STRUCTURE ALLOCATED IN CF(S) NAMED: FACIL04
CFNAME      STATUS/FAILURE REASON
-----
FACIL03     PREFERRED CF 1
                                     INFO110: 00000003 CC000B00 00010010
FACIL04     PREFERRED CF ALREADY SELECTED
                                     INFO110: 00000003 CC000B00 00020011

1 REALLOCATE STEP(S): REBUILD

STRNAME: IRRXCF00_P001                            INDEX: 68
SIMPLEX STRUCTURE ALLOCATED IN CF(S) NAMED: FACIL03
CFNAME      STATUS/FAILURE REASON
-----
FACIL04     PREFERRED CF 1
                                     INFO110: 00000003 CC007800 00020011
FACIL03     PREFERRED CF ALREADY SELECTED
                                     INFO110: 00000003 CC007000 00010010

1 REALLOCATE STEP(S): REBUILD

```

## Report example

```

STRNAME: ISTGENERIC                                INDEX: 15
CFNAME      STATUS/FAILURE REASON
-----
FACIL04     PREFERRED CF 1
                                     INFO110: 00000003 CC000B00 00000011
FACIL03     PREFERRED CF ALREADY SELECTED
                                     INFO110: 00000003 CC000B00 00000010

STRNAME: IXC_BIG_1                                INDEX: 90
ACTIVE POLICY INFORMATION USED BUT EXCLUSION LIST WAS IGNORED.
CFNAME      STATUS/FAILURE REASON
-----
FACIL03     PREFERRED CF 1
                                     INFO110: 00000003 CC007800 00000010
FACIL04     PREFERRED CF ALREADY SELECTED
                                     INFO110: 00000003 CC007800 00000011

```

## Report example

```

CFNAME: FACIL03
COUPLING FACILITY      : 002097.IBM.02.00000001DE50
                        PARTITION: 0E  CPCID: 00

CONNECTED SYSTEM(S):
#@$A      #@$2      #@$3

ACTIVE STRUCTURE(S):
DB8QU_SCA      DFHXQLS_#@$STOR1      IGWLOCK00
IM0A_EMHP      IM0A_IRLM      IM0A_MSGP
IRRXCFO0_B001  ISGLOCK      IXC_BIG_1
IXC_DEFAULT_1  JES2CKPT_1      LOG_IGWSHUNT_001
RRS_ARCHIVE_1  RRS_DELAYEDUR_1      RRS_MAINUR_1
RRS_RESTART_1  RRS_RMDATA_1      SYSIGGCAS_ECS
SYSZWLM_DE502097  SYSZWLM_WORKUNIT

```

## Report example

```

-----
REALLOCATE TEST RESULTED IN THE FOLLOWING:
  4 STRUCTURE(S) REALLOCATED - SIMPLEX
  0 STRUCTURE(S) REALLOCATED - DUPLEXED
  0 STRUCTURE(S) POLICY CHANGE MADE - SIMPLEX
  0 STRUCTURE(S) POLICY CHANGE MADE - DUPLEXED
 23 STRUCTURE(S) ALREADY ALLOCATED IN PREFERRED CF - SIMPLEX
  0 STRUCTURE(S) ALREADY ALLOCATED IN PREFERRED CF - DUPLEXED
  0 STRUCTURE(S) NOT PROCESSED
 357 STRUCTURE(S) NOT ALLOCATED
 128 STRUCTURE(S) NOT DEFINED
-----
 512 TOTAL

  0 STRUCTURE(S) WITH AN ERROR/EXCEPTION CONDITION

```

## Migration and coexistence

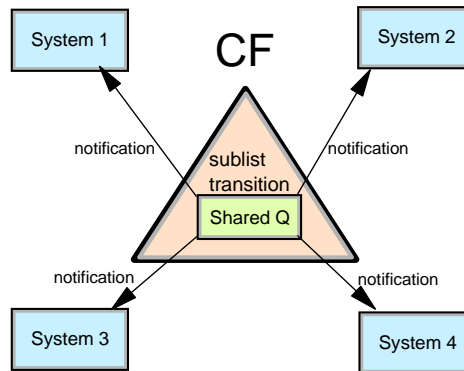
The enhancements to REALLOCATE are only available in z/OS V1R12

But other systems in the Parallel Sysplex running z/OS V1R10 or z/OS V1R11 need to install the PTF for OA29236 for this enhancement to work in z/OS V1R12

### New message

- IXC347I will display the results of a REALLOCATE REPORT and TEST

## Sublist notification



CF Keyed list structures  
used by IMS and MQ shared Queues

## Sublist notification delay enhancement

Sublist notification delay was introduced in z/OS V1R10 with CF Level 16 to reduce the overhead of telling all interested parties, when only one of them will actually be able to retrieve the message. It achieves this by:

- Send notification to one system (in round robin) to handle the work first
- Give it 5 milliseconds to collect the message (sublist notification delay)
- If message not collected, inform other systems

This has the effect of moving queue sharing from a pull-based workload balancing to more of a round-robin model

## Sublist notification delay enhancement

However, some customers had "hot standby" queue server address spaces and work would pause when that server would be the first to be notified.

To address this, the sublist notification delay was externalized, on a structure-by-structure basis, to let customers influence whether the load balancing is more inclined towards round-robin or pull based:

- New keyword `SUBNOTIFYDELAY` on the `STRUCTURE` definitions
- Valid values are from 0 to 1000000 (microseconds)
- The new value is shown in the output from the `D XCF,STRUCTURE,STRNM=xyz` command

## Display structure info (partial)

```
D XCF,STR,STRNM=IM0A_MSGP
IXC360I 18.29.21 DISPLAY XCF 668
STRNAME: IM0A_MSGP
STATUS: ALLOCATED
EVENT MANAGEMENT: MESSAGE-BASED
TYPE: SERIALIZED LIST
POLICY INFORMATION:
POLICY SIZE      : 24000 K
POLICY INITSIZE  : 15872 K
POLICY MINSIZE   : 10000 K
FULLTHRESHOLD   : 80
ALLOWAUTOALT    : NO
REBUILD PERCENT : N/A
DUPLEX          : ALLOWED
ALLOWREALLOCATE : YES
PREFERENCE LIST : FACIL03  FACIL04
ENFORCEORDER    : NO
EXCLUSION LIST  IS EMPTY
SUBNOTIFYDELAY  : 4000
.....
ACTUAL SUBNOTIFYDELAY: 4000
```



## Migration and coexistence

Requires CF level 16

Enhancement is rolled back to z/OS V1R9, z/OS V1R10, and z/OS V1R11 with APAR OA30994

PTF for OA30994 (or z/OS V1R12) must be installed on all systems in the sysplex before you start using **SUBNOTIFYDELAY** on the **STRUCTURE** definition in the **CFRM** policy.

Specifying the parm is optional and the default is 5 milliseconds.



New **FUNCTIONS** functions!



**Redbooks Workshop**

IBM ITSO - International Technical Support Organization

## Granular control of sysplex functions

z/OS V1R10 introduced the ability to control some sysplex related functions can be enabled at a system level.

The **COUPLExx** member gives the ability to **ENABLE** or **DISABLE** them at IPL time:

- **FUNCTIONS ENABLE(name1,name2) or DISABLE(name3)**

The **SETXCF** command lets you dynamically change most of them:

- **SETXCF FUNCTIONS,ENABLE=(name1,name2) or DISABLE(name3)**

The names of (nearly) all the functions can be found in a table in "Setting up a Sysplex" manual.

New in z/OS V1R12 are the functions **CRITICALPAGING** and **DUPLEXCFDIAG** - BOTH ARE DISABLED BY DEFAULT

## Function CRITICALPAGING

Some customers have experienced problems with HyperSwap because critical pages were paged-out and could not be paged-in once the HyperSwap had started:

- No I/Os (either reads or writes) are permitted once the HyperSwap process starts

## Function CRITICALPAGING

To address this, a two phase approach is taken:

- The IBM-provided default Program Properties Table (PPT) has been updated to define XCF, the HyperSwap programs, and GDPS programs as CRITICALPAGING:
  - This tells the system that, if at all possible, pages belonging to these address spaces should not get paged-out.
  - In the SCHEDxx member, you can specify CRITICALPAGING for other programs, through a new parameter on each PPT entry. However you CANNOT override the CRITICALPAGING setting in the default IBM supplied PPT in the SYS1.LINKLIB(IEFSDPPT). A table in the Initialization and Tuning manual shows the default PPT settings.
- A new XCF FUNCTION called CRITICALPAGING has been introduced:
  - This lets you control, at the system level, whether the CRITICALPAGING specification is heeded or not - by default, it is NOT.

## Function CRITICALPAGING

So what do I have to do if I use HyperSwap? By default, this new capability is DISABLED, so you need to update COUPLExx to say:

- FUNCTIONS ENABLE(CRITICALPAGING)

Note that you CANNOT change the setting of CRITICALPAGING using the SETXCF FUNCTIONS command - this can only be changed using an IPL:

- If a program is changed from NOCRITICALPAGING to CRITICALPAGING, any pages for that program that are already paged out will NOT be automatically paged in as a result of the change

## Function CRITICALPAGING

You can display the current setting of the **CRITICALPAGING** function using the **D XCF,C** command:

```
D XCF,C
IXC357I 10.47.25 DISPLAY XCF 719
SYSTEM #@$3 DATA
  INTERVAL  OPNOTIFY  MAXMSG  CLEANUP  RETRY  CLASSLEN
    165      168      2000     15      10      956

  SSUM ACTION  SSUM INTERVAL  SSUM LIMIT  WEIGHT  MEMSTALLTIME
    ISOLATE           0          NONE      N/A      N/A

CFSTRHANGTIME
  N/A
...

OPTIONAL FUNCTION STATUS:
FUNCTION NAME      STATUS  DEFAULT
DUPLXCF16         DISABLED  DISABLED
SYSSTATDETECT     ENABLED  ENABLED
USERINTERVAL      DISABLED  DISABLED
CRITICALPAGING  DISABLED  DISABLED
DUPLXCFDIAG      DISABLED  DISABLED
```

Note that the default is **DISABLED**

## Function CRITICALPAGING

Both aspects of this new capability (the PPT support, and the XCF switch to enable or disable it) are rolled back to z/OS V1R10 and V1R11 with APARs OA31691, OA31707, and OA31331.

The doc for this new function is in APAR OA31707 and in the V1R12 Setting Up a Sysplex manual.

## Function DUPLEXCFDIAG: the history

The debugging of CFCC-related problems is sometimes impeded by the inability to capture diagnostic information from the CF.

There are two kind of CF dumps:

- Soft CF dumps were unserialized (CF activity is not quiesced), so these result in an inconsistent picture and therefore may not be useful for diagnostic purposes
- Disruptive dumps. These result in a CF failure and all structures in the CF will need to be recovered

Collecting information for problems like break-duplexing previously required the installation of special diagnostic CFCC code, which would result in a disruptive dump being taken

## Function DUPLEXCFDIAG

CFLEVEL 17 introduced non-disruptive CF dump capability enhancements

z/OS APARs OA31387 and OA31392 added z/OS support (back to z/OS V1R10) to use this capability to initiate a non-disruptive dump in case of certain CF problems. The structure will be marked as "failed", rather than terminating the whole CF as would happen with a disruptive dump

## Function DUPLEXCFDIAG

There are various ways that a dump can be triggered:

- CF-related problems, such as a break-duplexing event
- Dumps initiated from the CF Console on the HMC
- Problems that are detected by a link

The first two always result in a dump being captured.

## Function DUPLEXCFDIAG

But link-detected problems result in dump requests being propagated across the link network **IF YOU PERMIT IT.**

The other part of APAR OA31387 is that it provides a new **FUNCTION** called **DUPLEXCFDIAG**.

This **FUNCTION** controls whether the **CF** links on *that* system are allowed to trigger one of these propagated dumps.

The default is that this capability is turned **OFF**.

## Function DUPLEXCFDIAG

This new function is currently only documented in the PTF Cover letter, or in APAR OA31387.

- The Setting Up a Sysplex manual will be updated

### New message:

- IXL051E information about the error, the structure, the CF dump that is initiated, and whether it is a disruptive dump or not

### Updated messages:

- IXC357I and IXC373I, will show the DUPLEXCFDIAG status

A PTF is available for z/OS V1R10 to z/OS V1R12

**NOTE:** It is recommended that DUPLEXCFDIAG only be enabled under the guidance of IBM Level 2

## New sysplex-related health checks

CF Processors

CF memory utilization

CF structure policy size

CDS max systems

CFRM message based check

SFM Structure hang time

## XCF\_CF\_PROCESSORS check

Check if the CF processor configuration is consistent with the IBM recommendation for this CFLEVEL (the recommendation is normally to use dedicated processors):

- CFs can be excluded from the check in the policy statement in HZSPRMxx (i.e. for test systems)

```
UPDATE CHECK(IBMxcf,XCF_CF_PROCESSORS)
SEVERITY(MED) INTERVAL(004:00) DATE (20090707)
PARAM(EXCLUDE(CFName, CFName, ...))
REASON('These are CFs in a test sysplex')
```

**NOTE:** This check is disabled in a VM environment



## XCF\_CF\_MEMORY\_UTILIZATION check

As the number of structures increases, and structure sizes grow over time (because of AUTOALTER, for example), it is possible for CFs to become short of storage:

- This check checks that the amount of memory allocated for structures and dump space is below a certain percentage.
- The percentage (1-99) can be specified in the policy statement in HZSPRMxx (default is 60%)

```
UPDATE CHECK(IBMxcf,XCF_CF_MEMORY_UTILIZATION)
SEVERITY(MED) INTERVAL(001:00) DATE (20090707)
PARM('MAXUTILIZATION(60)')
REASON('Coupling facility memory should not be over utilized')
```

## XCF\_CF\_MEMORY\_UTILIZATION check

Note that at this time, the check does not make any allowance for duplexed structures.

If the bulk of your CF storage is used by duplexed structures, you can set the threshold on the high side. If you do not have ANY duplexed structures, set the threshold to be <45%.

**NOTE:** At 90% storage utilization, XCF will automatically contract structure sizes in an attempt to free up CF storage to allow additional structures to be allocated

## XCF\_CF\_STR\_POLICYSIZE check

When allocating a structure, the CF creates enough control blocks to support the maximum size defined for that structure - these controls blocks come out of the space defined for the structure.

A huge difference between INITSIZE and SIZE could leave little or no usable structure space, resulting in allocation problems.

- Check that INITSIZE is at least half of SIZE

For structures that do not support ALTER, specifying different SIZE and INITSIZE values is a waste of CF storage

- Check that INITSIZE is equal to SIZE for structures that do not support ALTER.

## XCF\_CDS\_MAXSYSTEM check

If a function CDS (i.e. LOGR, WLM, OMVS) has a smaller MAXSYSTEM value than the Sysplex CDS, system MAXSYSTEM+1 (as specified in Sysplex CDS) would be allowed to join the sysplex, but not be permitted to use the function associated with that CDS.

- Check that the MAXSYSTEM value in all function CDSes is at least as large as the MAXSYSTEM value in the Sysplex Couple Data Sets. If not, raise an exception

## XCF\_CFRM\_MSGBASED check

The CFRM Message-Based event management protocol can significantly reduce recovery time for structures with large numbers of connectors when CF recovery is required.

- When active, it is used for all structures except signaling

CFRM Couple Data Sets should be formatted with:

- ITEM NAME(MSGBASED) NUMBER(1)

Check if the CFRM is enabled for message-based event management. If not, raise an exception

## XCF\_SFM\_CFSTRHANGTIME

Structure connectors should respond in a timely fashion to structure-related events to prevent sympathy sickness.

Sysplex Failure Management in z/OS 1.12 provides the ability to take automatic action on such cases. This is activated by the CFSTRHANGTIME keyword in the SFM policy.

If CFSTRHANGTIME(NO) (the default) is specified, this new function will not be active

This check checks the CFSTRHANGTIME value in the SFM policy and compares it to the recommended value.

- The current recommendation is 300 seconds



International Technical Support Organization

[ibm.com/redbooks](http://ibm.com/redbooks)

## System Logger enhancements



© 2010 IBM Corporation. All rights reserved.

[ibm.com/redbooks](http://ibm.com/redbooks)

International Technical Support Organization



## System logger enhancements in z/OS 1.12

**Automatic adjustment for incorrect SHAREOPTIONS on new  
Logger data set allocations**

**LISTCAT keyword for IXCMIAPU utility**

**4GB Logstream data set support**

**Virtual Storage Constraint Relief**



©2010 IBM Corporation. All rights reserved.

## System Logger SHAREOPTIONS enhancement

System Logger requires that multiple systems in a sysplex can access and update offload and staging data sets, which means that data sets must be allocated with SHAREOPTIONS(3 3).

Improper SHAREOPTIONS can lead to access errors, and an outage of the connector.

To get the correct SHAREOPTIONS, use an appropriate SMS data class:

- If no SHAREOPTIONS(3,3) are specified in the data class, the default SHAREOPTIONS of (1 3) is used
- For Offload data sets, CISIZE of 24KB is recommended
- For Staging data set, CISIZE of 4KB MUST be specified
  - So, you need TWO data classes for Logger data sets - both should specify SHAREOPTIONS(3 3)..

## System Logger SHAREOPTIONS enhancement

Prior to z/OS 1.12, warning messages were issued when Logger encountered a data set with the wrong SHAREOPTIONS

Starting with 1.12, System Logger automatically corrects wrong SHAREOPTIONS for NEW log stream data sets, through an IDCAMS ALTER:

- Messages are issued to indicate the problem and the automatic corrective action taken.
- The objective is to prevent data set access problems when SHAREOPTIONS(3 3) are not used.
- This does NOT imply that you do not need to correct the SMS DATACLASS settings

Applicable to new offload and staging data sets

- Will NOT change SHAREOPTIONS of existing data sets

## Usage considerations

### New and existing messages

- New IXG282I Corrections were made by Logger
- Existing IXG267I
- Existing IXG268I
- Existing IXG269I

**Review your automation policy for the existing messages**

## Gathering information about Logger data sets

One of the challenges for IBM when trying to diagnose Logger-related problems is that we have traditionally had very limited information about the Logger data sets.

- Remember that IXCMIAPU just feeds from LOGR CDS

To help address this, z/OS 1.12 adds the ability to include a LISTCAT when you run an IXCMIAPU LIST for a log stream:

- LIST LOGSTREAM NAME(logstreamname) DETAIL(YES) LISTCAT

This adds LISTCAT output for the data sets and helps you see the exact data set attributes:

- Be careful when using wildcards in logstream names together with the LISTCAT option because of the volume of output that might result, plus the traffic this can generate against the catalog

## Example of the LIST LOGSTREAM output

```

DATA SET NAMES IN USE: IXGLOGR.STC.DSTCPA21.DFHLOG.<SEQ#>

Ext.   <SEQ#>   Lowest Blockid /   Highest GMT /   Highest Local /
        Highest Blockid   Highest RBA       System Name
-----
*00001  A0000000  0000000000253F65  04/23/09 12:43:50  04/23/09 08:43:5
        00000000007A832F  0055446A

/* IDCAMS COMMAND */
LISTCAT ENTRIES( IXGLOGR.STC.DSTCPA21.DFHLOG.A0000000 ) ALL
CLUSTER ----- IXGLOGR.STC.DSTCPA21.DFHLOG.A0000000
IN-CAT --- UCAT.V#@$#M1
HISTORY
DATASET-OWNER----- (NULL)      CREATION-----2009.082
RELEASE-----2      EXPIRATION-----0000.000
SMSDATA
STORAGECLASS ---#@LSK110      MANAGEMENTCLASS--- (NULL)
----> DATACLASS ----- (NULL)      LBACKUP ---0000.000.0000
CA-RECLAIM----- (YES)
.....
ATTRIBUTES
KEYLEN-----0      AVGLRECL-----0      BUFSPACE-----
RKP-----0      MAXLRECL-----0      EXCPEXIT-----
----> SHROPTNS ( 3, 3 )      RECOVERY      UNIQUE      NOERASE      LINEAR

```

## Log stream data set 4GB support

Logger exploiters that generate large volumes of log data may cause frequent offload data set switches:

- Each data set switch impacts performance and has the risk of delay
- Can cause data set extents in LOGR CDS to fill quickly (by default, you can have up to 168 offload data sets per log stream)

The limit (of 2GB) on staging data set size:

- May cause more frequent offloading than desired
- For CF log streams, may lead to under-utilization of structure space

Growing workloads and increasing use of Logger have resulted in requests for larger Logger data set sizes.

- Limit prior to z/OS 1.12 was 2GB for both Offload and Staging data sets



## Log stream data set 4GB support

**z/OS 1.12 increased maximum log stream (staging AND offload) data set size from 2GB to 4GB**

**Logger data set sizes are specified (in units of 4KB) in the log stream definition:**

**- Offload data set parameters:**

- LS\_SIZE()
- LS\_DATACLAS() <===== THIS is the recommended way

**- Staging data set parameters**

- STG\_SIZE()
- STG\_DATACLAS() <===== THIS is the recommended way

## Log stream data set 4GB support

**Prior to 4GB data set support, if you specified a SIZE value that would have resulted in a >2GB data set, the value would be ignored and you would be given a 2GB data set.**



## Log stream data set 4GB support

**AFTER** the 4GB support is added, whatever size you provide will be used.

- If you specify a size that would result in a data set size > 4GB, you get this:

```
$HASP373 LOGR4GB  STARTED - INIT 1      - CLASS A - SYS #@3
IEF196I IEF237I DA24 ALLOCATED TO SYS00950
IEF196I IEF285I   SYS10246.T120456.RA000.IEESYSAS.R0300087      KEPT
IEF196I IEF285I   VOL SER NOS=  DISTSU.
IXG251I IKJ56893I DATA SET IXGLOGR.BERTHW.TEST4GB.A0000000 NOT
ALLOCATED+
IXG251I IGD17103I CATALOG ERROR WHILE DEFINING VSAM DATA SET
IXG251I IXGLOGR.BERTHW.TEST4GB.A0000000
IXG251I RETURN CODE IS 140 REASON CODE IS 110 IGG0CLEV
IXG251I IGD306I UNEXPECTED ERROR DURING IGG0CLEV PROCESSING
IXG251I RETURN CODE 140 REASON CODE 110
..
IXG251I SYMPTOM RECORD CREATED, PROBLEM ID IS IGD00003
IXG251I IGD17219I UNABLE TO CONTINUE DEFINE OF DATA SET
IXG251I IXGLOGR.BERTHW.TEST4GB.A0000000
$HASP395 LOGR4GB  ENDED
```

## Log stream data set 4GB support

**Because VSAM allocations depend on calculations based on CI Size, CA Size, SMS default device type, SMS bytes/track value, and track size of the device that is actually used....**

- Setting Up a Sysplex "testing log data set parameter modifications" shows the process to follow to find the correct values in your situation.
- Below are the values used in an IBM test environment with all 3390 devices and a bytes/track value of 56664

CI Size	xx_SIZE parameter (in number of 4K)	resulting DS Size in bytes
Offload data set 4K	1048400	4,294,656,000
24K	1048400	4,294,656,000
Staging data set 4k	1048400	4,294,656,000

## Log stream data set 4GB support

If you want to test the effect of the values you specify:

- When you DEFINE a log stream, the first offload data set will be allocated as part of the IXCMIAPU job. Watch the IXG283I message to see what size you get:
  - If you UPDATE an existing log stream, the new value is not used OR CHECKED until the next time an offload causes a new data set to be allocated.

## Log stream data set 4GB support

```
DATA TYPE(LOGR) REPORT(YES)
DEFINE LOGSTREAM NAME(BERTHW.TEST4GB)
    STRUCTNAME(BERTHW_TEST4GB)
    LS_DATACLAS(LOGR24K)
    LS_SIZE(1048500)
    STG_DATACLAS(LOGR4K)
    STG_SIZE(1048500)
    STG_DUPLEX(YES) DUPLEXMODE(UNCOND)
    LOWOFFLOAD(75) HIGHOFFLOAD(80)
```

selected lines out of the syslog:

```
$HASP373 LOGR4GB STARTED - INIT 1 - CLASS A - SYS #@$3
IEF196I IGD101I SMS ALLOCATED TO DDNAME (SYS00939)
IEF196I          DSN (IXGLOGR.BERTHW.TEST4GB.A0000000          )
IEF196I          STORCLAS (#@LSK110) MGMTCLAS (          ) DATACLAS
IEF196I (LOGR24K)
IEF196I          VOL SER NOS FOR DATA COMPONENT= DISTF8
IXG283I OFFLOAD DATASET IXGLOGR.BERTHW.TEST4GB.A0000000 257
ALLOCATED NEW FOR LOGSTREAM BERTHW.TEST4GB
CISIZE=24K, SIZE=4294656000
$HASP395 LOGR4GB ENDED
```

## Log stream data set 4GB support

When you define a log stream, the staging data set is not allocated until someone connects to the log stream. If you specified a too-large value, the allocation will fail..

## Log stream data set 4GB support

Use **IXGCONLS** sample from the **SYS1.SAMPLIB** to force staging data set allocation

```
S IXGCONLS,LOGSTRM=BERTHW.TEST4GB
....
IXC582I STRUCTURE BERTHW_TEST4GB ALLOCATED BY SIZE/RATIOS. 330
  PHYSICAL STRUCTURE VERSION: C686D316 DBD21584
  STRUCTURE TYPE:             LIST
  CFNAME:                     FACIL04
....
IEF196I IGD101I SMS ALLOCATED TO DDNAME (SYS00945)
IEF196I          DSN (IXGLOGR.BERTHW.TEST4GB.##$3          )
IEF196I          STORCLAS (@LSK110) MGMTCLAS (          ) DATACLAS
IEF196I (LOGR4K)
IEF196I          VOL SER NOS FOR DATA COMPONENT= DISTF8
IXG283I STAGING DATASET IXGLOGR.BERTHW.TEST4GB.##$3 338
ALLOCATED NEW FOR LOGSTREAM BERTHW.TEST4GB
CISIZE=4K, SIZE=4294656000
IEF196I IGD103I SMS ALLOCATED TO DDNAME SYS00946
+IXG273I LOGSTREAM CONNECT COMPLETED SUCCESSFULLY
```

## Log stream data set 4GB support

### New messages

- IXG283I allocation CI and Size information
- IXG284I delete and pending delete

### APAR OA31461 (PTF UA52327, UA52328, UA52329)

- Must be applied on all active pre-z/OS V1R12 systems, before IPLing z/OS V1R12 in the sysplex.

### APAR OA30548 (PTF UA52443, UA52444, UA52445)

- Rolls back support for 4GB Logger data sets to z/OS V1R9, V1R10 and V1R11

## Logger problem avoidance

One of the challenges that System Logger users face is that problems related to offload data sets may only become visible when it is too late:

- Run out of DSEXTENTS because offload data sets are too small
- Run out of DSEXTENTS because offload data sets are being allocated at an accelerated rate

Unless you do an ISPF 3.4 for the offload data sets, the first indicator of a problem could be an abend in the connector

## Logger problem avoidance

One of the side-benefits of the IXG283I message added as part of the 4GB data set support, is that it gives you more visibility into what is going on, especially if offload data sets are being created very frequently.

```
IXG283I OFFLOAD DATASET IXGLOGR.IFASMF.STRIPE.TYPDFLT.A0000109
ALLOCATED NEW FOR LOGSTREAM IFASMF.STRIPE.TYPDFLT
CISIZE=24K, SIZE=491520000
```

Annotations:

- DSN (points to IXGLOGR.IFASMF.STRIPE.TYPDFLT.A0000109)
- Log stream name (points to IFASMF.STRIPE.TYPDFLT)
- Data set size (in bytes) (points to SIZE=491520000)
- CISize (should be 24K for offload data sets) (points to CISIZE=24K)

## VSCR improvements

Before z/OS V1R12, almost all code of System Logger was located in a single load module in ELPA

With z/OS V1R12, the System Logger code has been rearranged and approximately 0.5MB of System Logger code has been moved out of EPLPA to the Extended Private area, saving approximately 0.5MB of common storage below the 2GB bar

## CF Sizer enhancements

**CFSizer is Web-based tool, intended to help you size CF structures.**

**Available at:**

- <http://www.ibm.com/systems/support/z/cfsizer/>

**CFSizer has recently been enhanced to provide:**

- More accurate calculations for IMS Operations Management (OM) Audit, Resource, and OSAM structures
- Provide sizes for IBM Session Manager Sysplex User structures
- Improved sizes for XCF signaling structures
- Sizes for InfoSphere™ Classic control and filter structures
- Improved sizes for DB2 SCA structures
- Various usability improvements to the CFSIZER Web pages



## z196 Parallel Sysplex Changes

The zEnterprise (or z196 or 2817) introduces a number of sysplex-related changes:

- Connectivity improvements (up to 80 coupling links)
- Connectivity improvements with 128 coupling CHPIDs per server
- Coupling Link options
- No ETR connectivity - STP CTN is mandatory
- New CF Level - CF Level 17
- Support for LPARs with both dedicated and shared engines removed
- z196 migration considerations

## More physical CF links

Large Parallel Sysplex configurations (>100K MIPS), configurations that host multiple Parallel Sysplexes and CFs on the same CPCs, and sysplexes that span large distances, are driving the need for more than 64 Coupling Links

To address this sysplex growth, z196 supports more physical CF links - up from 64 to 80. This represents a full configuration of 32 PSIFB and 48 ISC-3 links

- The number of internal ICP CHPIDs remains at 32 for coupling between images on the same server

## More CF Link CHPIDs

Consider....

That the largest bandwidth currently available on FICON is 8 Gb/sec.

And you can have 16384 devices on a single FICON channel  
AND you can use PAV or HyperPAV to conceptually have more than 16384 concurrent I/Os running on a FICON channel...

12x Infiniband has a theoretical bandwidth of 60 Gb/sec

- Even old ISC links have a bandwidth of 2 Gb/sec

Each CF link CHPID currently supports 7 subchannels.

AND there is no PAV or HyperPAV for CFs

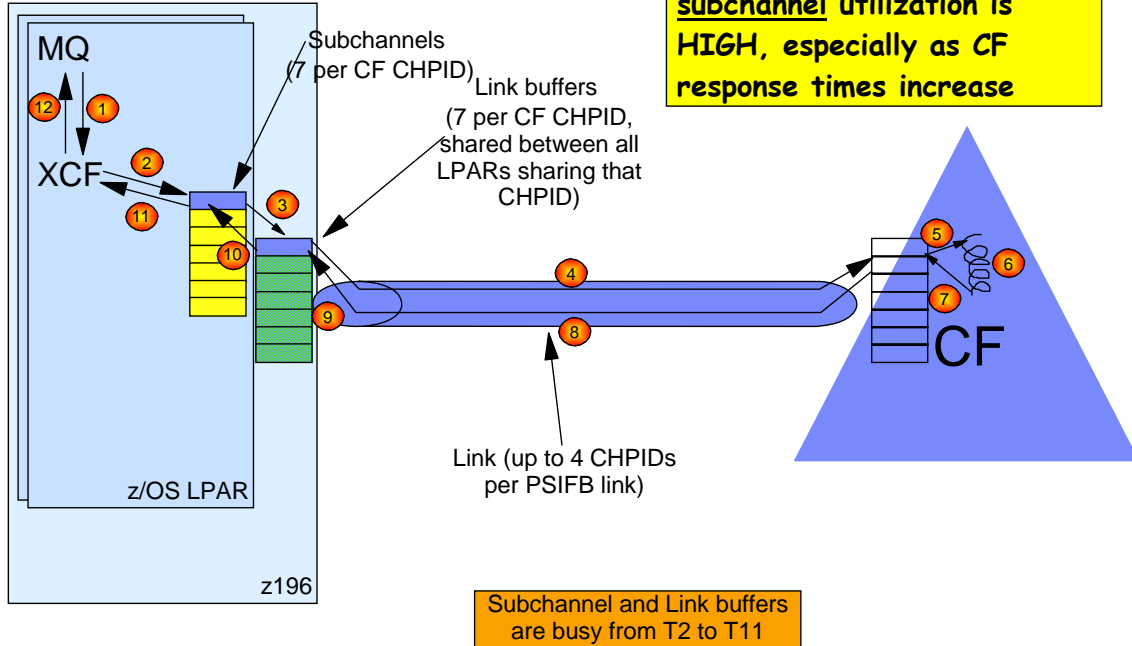
## More CF Link CHPIDs

So, what is the actual utilization of a *CF link* likely to be, compared to a FICON channel?





## More CF Link CHPIDs



## More CF Link CHPIDs

Another challenge, given that there is a finite number of CF links, is that you can't share CF Link CHPIDs between sysplexes. So, if you have three sysplexes spanning two CPCs, that used to mean a minimum of 6 CF links:

- And we have many customers with a lot more than three sysplexes
- And many sysplexes that require more than 2 links between their z/OS systems and their CFs

## More CF Link CHPIDs

To enable higher utilization of the physical link, AND, let you share one physical link between more than one sysplex, PSIFB links support sharing a link between up to (normally) 4 CHPIDs

- This effectively lets you have 28 subchannels per link instead of 7
- Or it lets you have 4 sysplexes all sharing one link

But, prior to z196, the limit on the number of CF link CHPIDs was the same as the maximum number of physical CF links - 64.

## More CF Link CHPIDs

In order to get more benefit from the ability to assign more than one CHPID to an PSIFB link, z196 increases the maximum number of CF link CHPIDs from 64 to 128

- This should allow to you drive more value from the same number of links
- Helps customers with large numbers of sysplexes sharing the same CPCs
- Provides better efficiencies for large distance sysplexes:
  - Fewer links (meaning fewer cards to purchase for the CPC) for the same number of subchannels
  - Means fewer DWDM ports than if you were using ISC links

**Side note - Late breaking news - there are now TWO DWDMs qualified for PSIFB - Adva and Cisco**

## z196 Coupling Link options

z196 supports a variety of CF link types that provide different distance and performance characteristics

Type	Description	Use	Theoretical Link rate	Distance	z196 maximum
ISC-3	InterSystem Channel-3	z196 to z196, z10, z9	2 Gbps	10 km un-repeated (6.2 miles) 100 km repeated	48
PSIFB	12x IB-DDR InfiniBand	z196 to z196, z10	60 Gbps	150 meters (492 feet)	32
	12x IB-SDR InfiniBand	z196 to z9	30 Gbps		
PSIFB LR	1x IB-DDR (b) InfiniBand	z196 to z196, z10	2.5 Gbps or 5 Gbps	10 km un-repeated (6.2 miles) 100 km repeated	32
IC	internal coupling channel	Internal communication	Internal speeds	N/A	32

b. Data rate (SDR or DDR) depends on the capabilities of the DWDM

**Note that there is NO ICB support**

## z196 Coupling Link options

### Other considerations:

- z196 does not support ICB links
- z196 is last generation that will support ordering of ISC links:
  - If you upgrade a z196 with ISC links to the next generation, you can carry the ISC links to that new box
  - But, the strategic direction appears to be clear
  - At the time of writing, 2 DWDMs are qualified for PSIFB links, but one more vendor is expected to be qualified by the end of 2010.

**The choice of the coupling link type has a critical effect on the response time and the overhead of using a CF**

## CF Overhead costs

Host CF	z890	z990	z9 BC	z9 EC	z10 BC	z10 EC	z196
z890 ISC	13%	15%	16%	17%	19%	21%	NA
z890 ICB	9%	10%	10%	11%	12%	13%	NA
z990 ISC	13%	14%	14%	15%	17%	19%	NA
z990 ICB	9%	9%	9%	10%	12%	13%	NA
z9 BC ISC	12%	13%	14%	15%	17%	19%	23%
z9 BC PSIFB 12X	NA	NA	NA	NA	13%	14%	16%
z9 BC ICB	8%	9%	9%	10%	11%	12%	NA
z9 EC ISC	12%	13%	13%	14%	16%	18%	22%
z9 EC PSIFB 12X	NA	NA	NA	NA	13%	14%	16%
z9 EC ICB	8%	8%	8%	9%	10%	11%	NA
z10 BC ISC	12%	13%	13%	14%	16%	18%	22%
z10 BC PSIFB 12X	NA	NA	11%	12%	13%	14%	15%
z10 BC ICB	8%	8%	8%	9%	10%	11%	NA
z10 EC ISC	11%	12%	12%	13%	15%	17%	22%
z10 EC PSIFB 12X	NA	NA	10%	11%	12%	13%	15%
z10 EC ICB	7%	7%	7%	8%	9%	10%	NA
z196 ISC	NA	NA	11%	12%	14%	16%	21%
z196 PSIFB 12X	NA	NA	9%	10%	11%	12%	14%

This is based on an "average" data sharing profile of 9 CF requests/MIPS/second  
 With z/OS 1.2 and above, synch->asynch conversion caps values in table at about 18%  
 PSIFB 1X links would fall approximately halfway between PSIFB 12X and ISC links  
 IC links scale with speed of host technology and would provide an 8% effect in each case

## STP

**z196 does not provide for direct connection to a Sysplex Timer**

**- Use of the STP protocol is mandatory on z196**

**z196 can be in the same STP CTN with z9 and z10 servers, but NOT with z990 or z890 servers**

**ISC-3 and InfiniBand links can be used for STP on a z196**

### NOTE:

**A Parallel Sysplex with ETR network must migrate to mixed CTN or STP-only CTN *before* introducing a z196**

**- In a mixed CTN, the System z196 may be a Stratum 2 or Stratum 3, but not a Stratum 1**

## New CFCC level - CF Level 17

z196 provides new CF Level - 17

New CF Level is NOT rolled back to earlier generations

More detail about this new CF Level is provided in the next section

## Migration and coexistence in a Parallel Sysplex

- System z196 does not support active participation of z900, z800, z990 and z890 servers in the same Parallel Sysplex
- z196 does not support ICB-4 coupling links
- Required CFCC levels on z9 and z10
- CF processor options
- STP requirements
- PSIFB connection to System z9
- z/OS service requirements

## Required CFCC levels

The following CFCC code levels are supported when coupling with a D/T2817 - z196

- D/T2094 - z9 EC - Driver\_67, Release 15.0 Service Level 2.11
- D/T2096 - z9 BC - Driver\_67, Release 15.0 Service Level 2.11
- D/T2097 - z10 EC - Driver\_79, Release 16.0 Service Level 2.25
- D/T2098 - z10 BC - Driver\_79, Release 16.0 Service Level 2.25

## CF processor options

Dynamic ICF expansion ("L-shaped LPARS") no longer supported on z196

- Dedicated ICF and Shared CP in the same CF are not supported
- Dedicated ICF and Shared ICF in the same CF are not supported

### Dedicated ICFs

- Always recommended for production

### Shared ICFs

- Second best, mainly intended for test sysplexes
- Review PR/SM Planning Guide for recommendations on weights, dynamic CF dispatching (DCFD), and capping
- Refer to techdoc "Parallel Sysplex Performance: Dynamic ICF Dispatching" TD102670

## PSIFB connection to System z9

ICB is not supported on z196

Last order date for PSIFB on System z9 has passed (June 30)

Only the 12x PSIFB links are supported, PSIFB-LR 1x connection to z9 is NOT supported

- Obviously you can use ISC3 links to connect z9 and z196.

PSIFB is not supported between multiple System z9, only to a System z10 or z196

Adding the first PSIFB adapter to a System z9 is disruptive, additional adapters can be added concurrently

## z/OS Service Requirements

Refer to the z196 PSP bucket for the *latest* list of z196-related service:

- Even better, use the FIXCATEGORY support in SMP/E

## Important documentation to review

### on InfiniBand:

- [SG24-7539 Getting Started with InfiniBand on System z10 and System z9](#)
- [SB10-7155 zEnterprise System PR/SM Planning Guide](#)
- [SG24-7833 IBM zEnterprise System Technical Guide](#)

### on CF processor options

- [SB10-7155 zEnterprise System PR/SM Planning Guide](#)
- [TD102670 Parallel Sysplex: Dynamic ICF dispatching](#)

### on STP

- [SG24-7281 Server Time Protocol Implementation Guide](#)
- [SAPR guide and STP checklist](#)
- [FLASH10672 Qualified WDMs for Server Time Protocol \(STP\)](#)





## CF Level 17

**CF Level 17 is delivered on z196 processors.**

- Only on z196 - not rolled back to earlier generations

**Delivers new capabilities and enhancements to existing functions:**

- Non-disruptive CF dumps
- Increased number of structures in a single CF
- Up to 255 CF list/lock structure connectors (was 32)
- Increased maximum structure size to 1 TB (up from 100GB)
- Support for up to 128 coupling CHPIDs (was 64)
- Remove physical coupling link limitation of 64 links. Instead, support whatever the machine physically can provide (was 64)
- Improved performance for structure alter
- Structure size changes

## CF Level 17

### Non-disruptive CF Dumps

**There are two types of CF-related dumps:**

- Structure dumps. These are taken under the control of z/OS, and the dump is written to the CF dump space (defined in the CFRM policy).
- CF dumps. These can be initiated from the CF Console, triggered by the CF itself in reaction to a problem, triggered by a link if it detects a problem, or initiated by z/OS. In all cases, the dump is written to a file on the HMC, NOT to the CF dumpspace.

## CF Level 17

Certain types of CF problems require a CF dump to gather the required information, however prior to CF Level 17, such dumps were disruptive - the whole CF would go down as part of the dump.

With CF Level 17, nearly all CF dumps can be taken non-disruptively. This allows IBM to gather the information we need to address CF-related problems, without impacting your sysplex availability.

## CF Level 17

### Non-disruptive CF Dumps

#### Non-disruptive dumps can be:

- Triggered automatically, based on certain events
  - Under direction of IBM support, you can tailor the triggers using the NDDUMP command on the CF console
- Initiated from z/OS
  - XCF will trigger a CF dump in response to certain types of errors.
- Initiated from the CF Console
  - Using the CFDUMP command.
  - This results in a dump being taken internally in the CF, and a message being issued to z/OS to inform it of the event.

## CF Level 17

**Increased number of structures allocated in a CF.**

**Prior to CF Level 17, you could not have more than 1023 structures in a CF.**

**For customers with many data sharing groups, or very large numbers of CICS regions that use CF log streams, this could be a constraint to growth in the sysplex**

- Remember that you need to allow for all structures residing in one CF in case of a failure.

**With CF Level 17, each CF can now contain up to 2047 allocated structures (was 1023 previously).**

## CF Level 17

**Considerations for CF failure - can't have 2047 structures in each CF, because if there is a failure, there would be more than 2047 structures in the surviving CF.**

**Unfortunately, the use of duplexed structures makes management and planning a little more complex:**

- If you have 1000 duplexed structures in CF01 and the second copy of those structures in CF02, if either CF fails, you will NOT carry over the duplexed copies (assuming you have just 2 CFs), so you need to count (separately) the number of simplex structures in each CF and the number of duplex ones, and plan accordingly.
- The increase in the limit from 1023 to 2047 should provide a lot of room for growth before you need to get this level of granularity in your planning.

## CF Level 17

In the early days of sysplex, the norm was 1 z/OS = 1 DB2. There is a max of 32 systems in a sysplex, so it was natural to assume a maximum of 32 DB2s in a data sharing group.

For various reasons, we are seeing multiple cases of many DB2 subsystems from the same data sharing group in the same z/OS system.

In support of this move to larger data sharing groups, CF Level 17 increases the number of connectors to a list or lock structure from 32 to up to 255:

- Lock structure now supports 247 connectors
- Serialized list structures supports 127 connectors
- Unserialized list structure supports 255 connectors
- Cache structures already support 255 connectors

## CF Level 17

### Increased maximum structure size

In order to feed data to complex applications as quickly as possible, you want to keep the data close to the CPU. In-z/OS buffers are the first choice, but in a data sharing environment, you need cache structures in the CF. And as applications consume more data, that drives up the cache sizes.

In addition, MQ queue sharing is becoming very popular, with more customers using this function, and more applications moving their private queues over to reside in the CF instead:

- While the queues would normally not hold a large number of messages, if there is a glitch, very large structures may be required to hold new messages until they start getting processed again.

## CF Level 17

Prior to CF Level 17, the largest supported structure size was 100GB.

- To put this in perspective, largest LPAR size on a z990 was only 256GB

With CF Level 17, structures can be defined to CFRM with **SIZE** or **INITSIZE** values of up to 1 TB.

## CF Level 17

**Support for more CHPIDs**

As discussed previously, z196 supports up to 80 CF Links. CF Level 17 increases the number of supported CHPIDs up to 128.

**Very important to remember that there is no longer a one-to-one correspondance between CF CHPIDs and CF links.**

- With PSIFB, you can have up to 4 CHPIDs per PSIFB port.
- For low utilization sysplexes (test or sandbox, for example) or sysplexes spread over very large distances, you can have up to 16 CHPIDs per HCA (2 PSIFB ports)

## CF Level 17

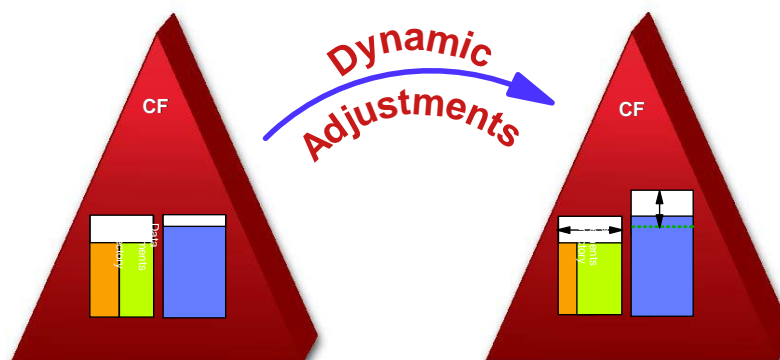
### Decreased structure Alter elapsed times

There are many valid reasons for altering an in-use CF structure:

- Structure might be too small for the current levels of usage
- Ratio between *defined* number of entries and elements is not in line with the *actual usage* of entries and elements
- Might need more record data entries in a lock structure
- Might need more space for event monitor controls control blocks

## CF Level 17

To remove some of the worry of sizing structures exactly "right", XES Auto Alter (enabled by ALLOWAUTOALTER in CFRM policy) can initiate alters automatically when specified thresholds are exceed.



## CF Level 17

The objective of doing an ALTER, rather than a REBUILD, is that it is supposed to take less time and be less disruptive than a REBUILD.

While ALTER does not quiesce activity to a structure the way REBUILD does, there have been cases where ALTERs can run for a long time, impacting performance during this time.

CF Level 17 implements improvements that eliminate most of the situations where ALTERs can run for extended times.

## CF Level 17

How would you know if Alters are taking a long time?

- High CF CPU utilization in RMF reports
- But number of CF requests is no higher than average
- Increased response times for CF requests

IF you see this, a possible cause is long-running Alters...

Check the console for Structure Alter messages:

- One at the start of the Alter process (IXC530I)
- And an Alter ended (IXC533I or IXC534I) message

If the time is significant, CF Level 17 should deliver noticeable benefits



## CF Level 17

### Free lunch?

Remember that some space is taken out of each structure for control blocks related to that structure.

So, as more functions and capabilities are added to CFCC, you typically see increases in the amount of space required for these control blocks - which **SHOULD** be reflected in increased structure sizes.

For CF Level 17, tests in IBM indicate an increase of 0-4% compared to CF Level 16.

To be safe, use CF Sizer.

## CF Level 17

### Required software levels:

- To *exploit* the new functions in CF Level 17, you must be running:
  - z/OS 1.10 or later with OA32807 or z/VM 5.4
- However, you can *connect* to CF Level 17, AS LONG AS YOU DON'T TRY TO USE ANY NEW FUNCTIONS, using z/OS 1.7, 1.8, or 1.9 with toleration service applied.
- For the list of APARs, refer to the 2817DEVICE PSP bucket.
- However, no matter what release you are on, we recommend pulling all the latest **FULL** Enhanced HOLDDATA (the one with Fix Category info) and then run:

```
REPORT
MISSINGFIX
ZONES (R12T100)
FIXCAT(IBM.Device.Server.z196-2817).
```





## CF Service Management

**z/OS generally delivers new functions via new releases (1.11, 1.12, and so on). And CF delivers new functions via new CF Levels (15, 16, 17, and so on).**

**And, just as z/OS can have bugs, equally, CFCC can also have bugs:**

- z/OS bugs are addressed by APARs and PTFs.
- CFCC bugs are addressed by licensed internal code (HW) fixes..

**However, CFCC is managed as part of the hardware service process, and z/OS service is managed by SMP/E.**

**How to control and coordinate the two....?**

## CF Service Management

The hardware people speak a different (strange!) language...  
What do they speak? And how do we see what they are referring to?

Hardware service is described in terms of:

- Driver levels (Driver 76, Driver 79, etc). Driver levels are generally associated with a "GA" level of a particular generation of CPC. A Driver is similar to software releases.
- Service Streams are similar in concept to components in z/OS (Logger, XCF, etc)
- Bundles (37, 24, etc) are conceptually similar to z/OS RSUs
- MCLs (N24403.006, N24403.007) are similar to PTFs. For CFCC, a new MCL level delivers a new Service Level

## CF Service Management

MCLs are the most granular level at which "service" can be applied to a CPC.

It is possible to apply an MCL to one service stream, but not to another:

- However, within a service stream, MCLs are "mandatory sequential" - that is, every MCL pre-reqs every other MCL in that stream. For example, you cannot install MCL N24403.004, skip over N24403.005, and install N24403.006. If you install N24403.006, you automatically install all previous levels as well.

While MCL numbers are sequential, CF Service Level names are *not* necessarily sequential - we might go from 2.16 directly to 2.20.

## CF Service Management

On the z/OS side, we don't talk about MCLs and such things. Using the D CF command (or the CF Operator Console), we see CF Service Levels:

```

D CF
IXL150I 17.44.06 DISPLAY CF 423
COUPLING FACILITY 002817.IBM.02.0000000B3BD5
PARTITION: 2F CPCID: 00
CONTROL UNIT ID: FFF2

NAMED FACIL04
COUPLING FACILITY SPACE UTILIZATION
ALLOCATED SPACE          DUMP SPACE UTILIZATION
STRUCTURES:              116 M      STRUCTURE DUMP TABLES:      0 M
DUMP SPACE:              2 M        TABLE COUNT:              0
FREE SPACE:              465 M      FREE DUMP SPACE:           30 M
TOTAL SPACE:             583 M      TOTAL DUMP SPACE:         30 M
                                MAX REQUESTED DUMP SPACE:      0 M
                                STORAGE INCREMENT SIZE:      1 M

VOLATILE:                YES
CFLEVEL:                 17

```

→ **CFCC RELEASE 17.00, SERVICE LEVEL 02.18**  
 BUILT ON 08/02/2010 AT 10:32:00  
 COUPLING FACILITY HAS 1 SHARED AND 0 DEDICATED PROCESSORS  
 DYNAMIC CF DISPATCHING: ON

## CF Service Level management

How do we publish information about CFCC bugs?

IBM released two APARs in 2010 describing CFCC problems - OA31960 and OA31604. These were intended to make customers aware of two CFCC problems, however this mechanism does not make sense in the long term:

- How do we get rid of them? If we ship a PTF, and you apply the PTF, reporting will go away, but if you don't apply the MCL, the problem still exists.
- Or, you could apply the MCL, so the problem is fixed, but if the PTF is not applied, this will still show up as an error on your system...

## CF Service Level management

**SOMETIMES** we also provide **CFCC** level information in **EC HOLDS** in **PTFS**:

```

++ HOLD(UA53157) SYS FMID(HBB7760) REASON(EC) DATE(10070)
COMMENT
(*****
 * FUNCTION AFFECTED: BCP                               (OA30438) *
 *                               XES                               *
*****
 * DESCRIPTION      : Engineering change                *
*****
 * TIMING           : Exploitation                       *
*****
This fix is not effective on a given system unless the list
structures accessed by that system reside on a coupling facility
with the new Set Monitored States command installed. The
applicable Coupling Facility Control Code microcode loads (MCLs)
are:

z10 (2097 / 2098)
DR76 CFCC EC N10964 MCL013 CFCC Release 16 Service Level 2.22
DR79 CFCC EC N24403 MCL004 CFCC Release 16 Service Level 2.22

z9 (2094 / 2096)
DR67 CFCC EC G40953 MCL014 CFCC Release 15 Service Level 2.11
    
```

Sometimes we need an MCL to be applied to activate a PTF, however there is no link between SMP and HW

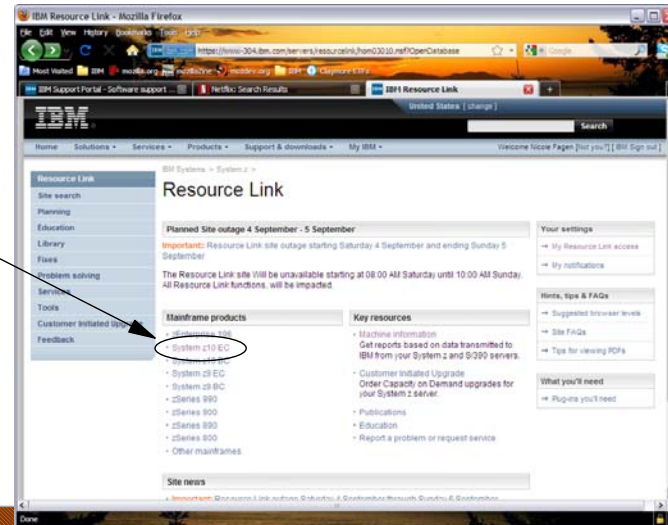
on. All rights reserved.

## CF Service Management

To find out information about the CF-related MCLs that are available, logon to ResourceLink:

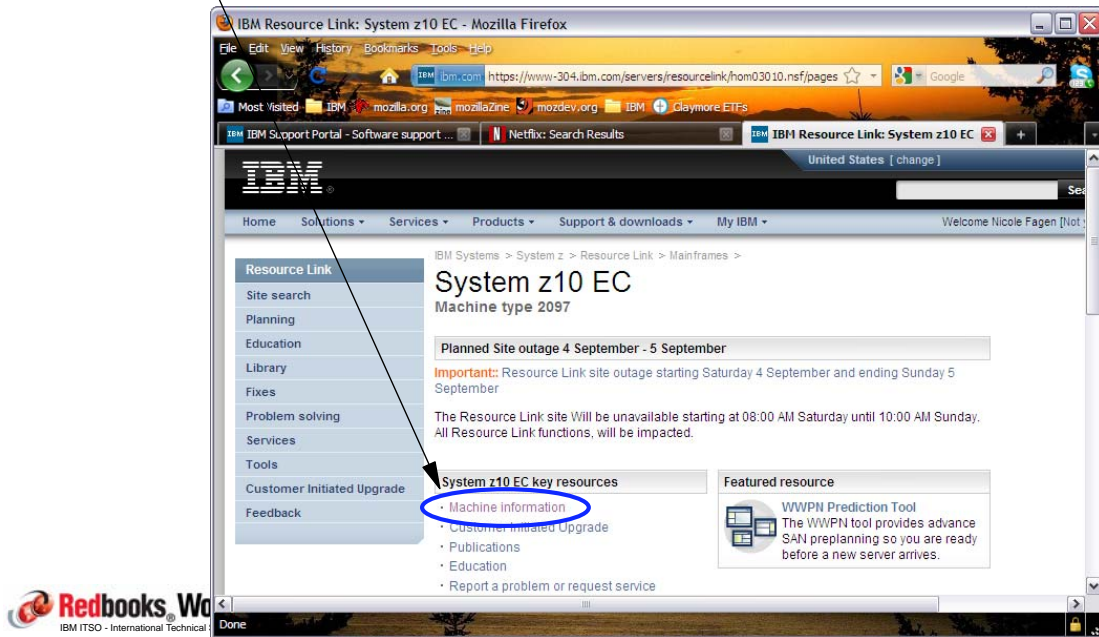
(<https://www-304.ibm.com/servers/resourcelink/svc03100.nsf?OpenDatabase>)

Then select the processor you are interested in



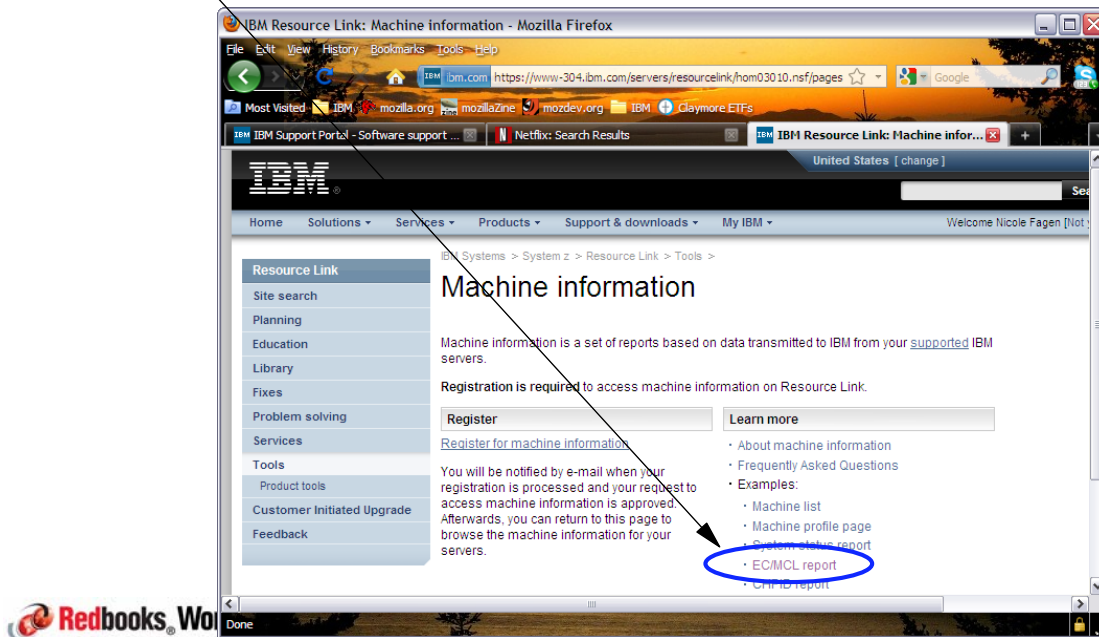
# CF Service Management

Then select Machine Information



# CF Service Management

Then select EC/MCL report





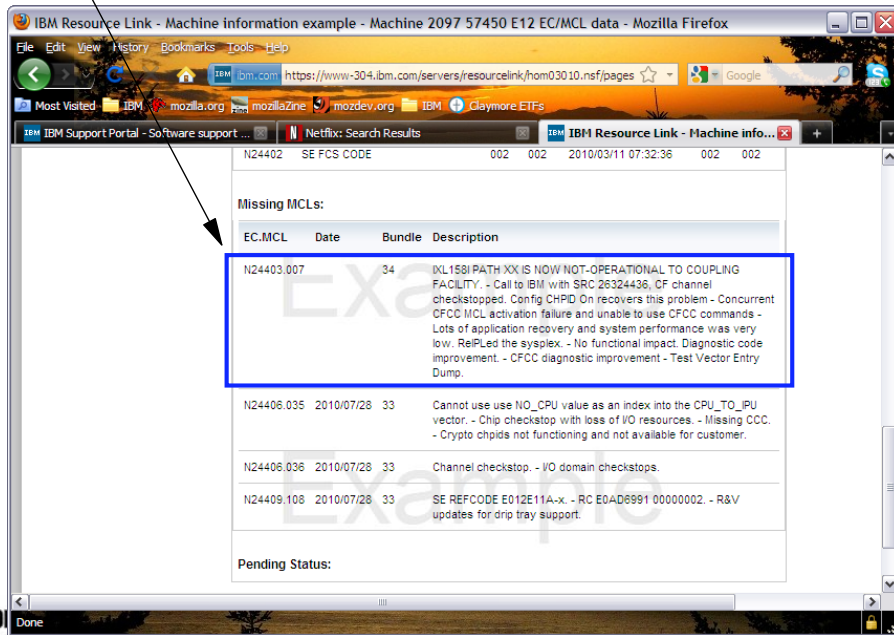
# CF Service Management

Finally, select ALL



# CF Service Management

And scroll down looking for any info in the CFCC stream....



## CF Service Management

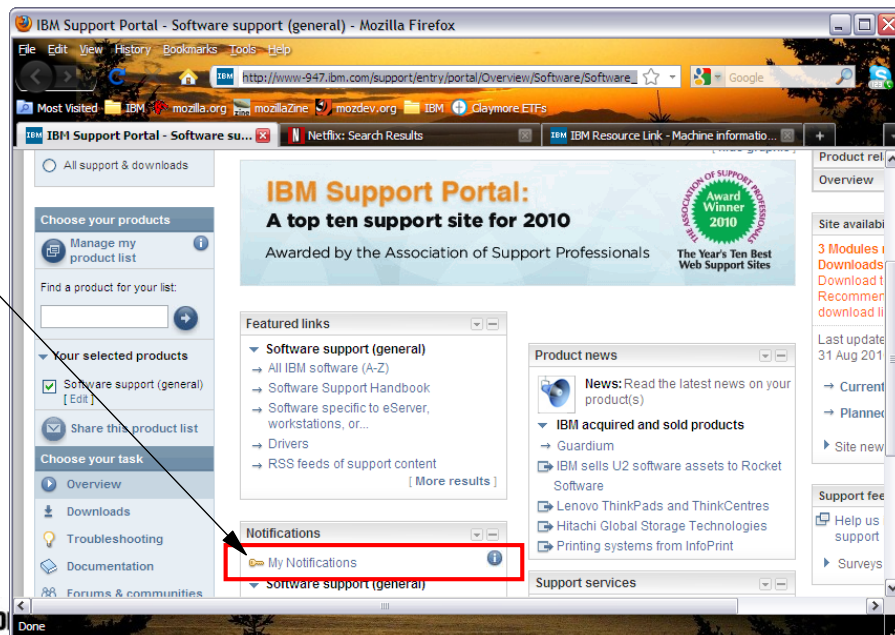
To get notified in case serious processor problems are discovered, subscribe using ResourceLink or the software support portal:

([http://www.ibm.com/support/entry/portal/Overview/Software/Software\\_support\\_%28general%29?pgel=wspace](http://www.ibm.com/support/entry/portal/Overview/Software/Software_support_%28general%29?pgel=wspace))

## CF Service Management

You will have to sign in using your IBM ID

Then select  
My  
Notifications

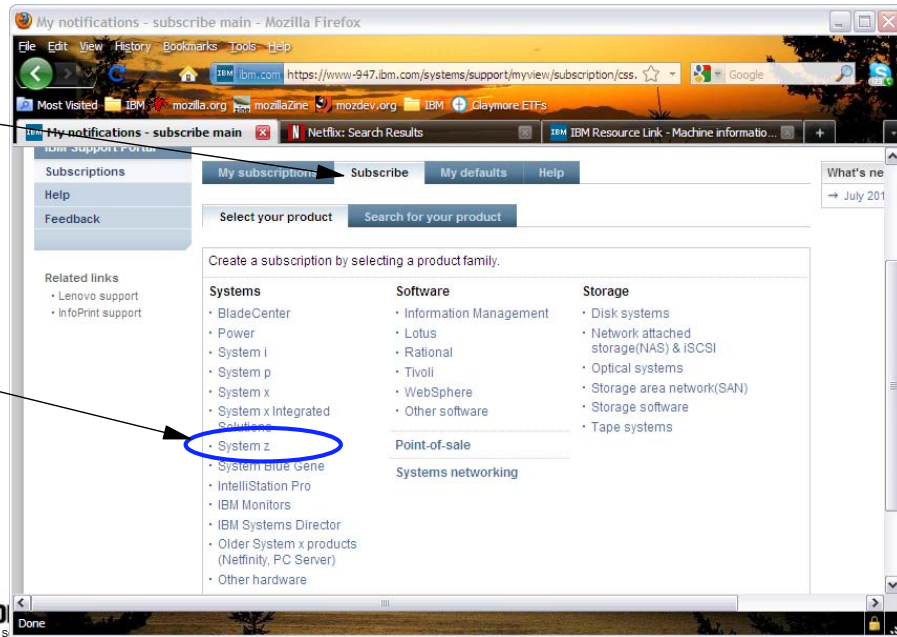


The screenshot shows the IBM Support Portal interface in a Mozilla Firefox browser window. The page title is "IBM Support Portal - Software support (general) - Mozilla Firefox". The URL in the address bar is "http://www-947.ibm.com/support/entry/portal/Overview/Software/Software...". The page content includes a navigation sidebar on the left with sections like "Choose your products", "Your selected products", "Choose your task", and "Notifications". The "Notifications" section is expanded, and the "My Notifications" link is highlighted with a red rectangular box. An arrow points from the text "Then select My Notifications" to this box. Other sections on the page include "Featured links", "Product news", and "Support services".

# CF Service Management

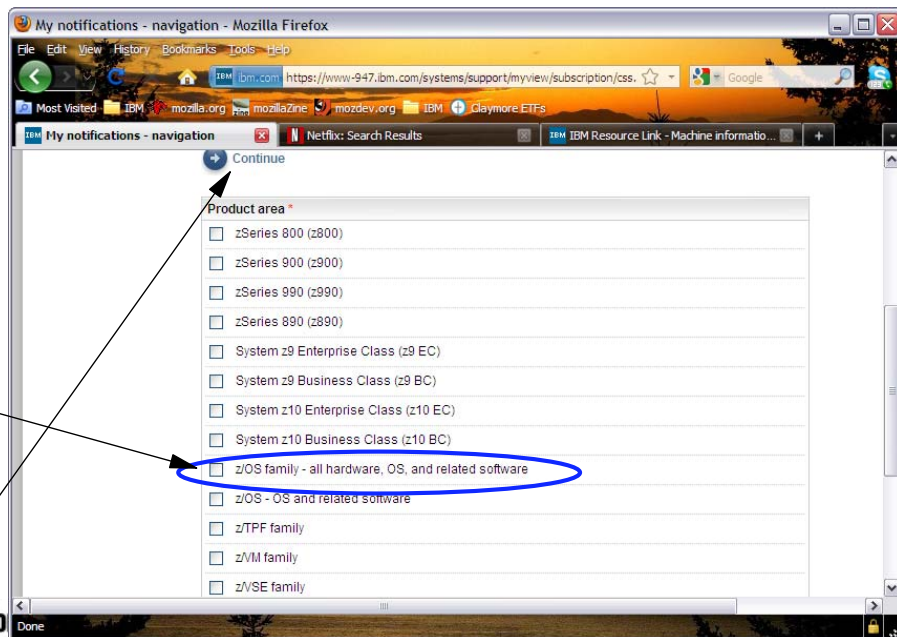
Select  
Subscribe  
tab

Then  
System z



# CF Service Management

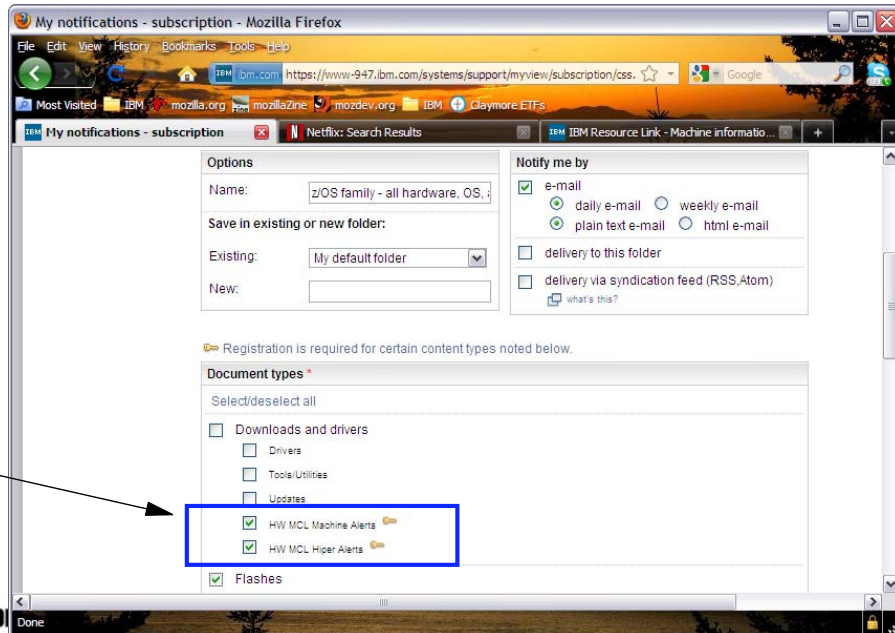
Then z/OS  
family  
and Continue





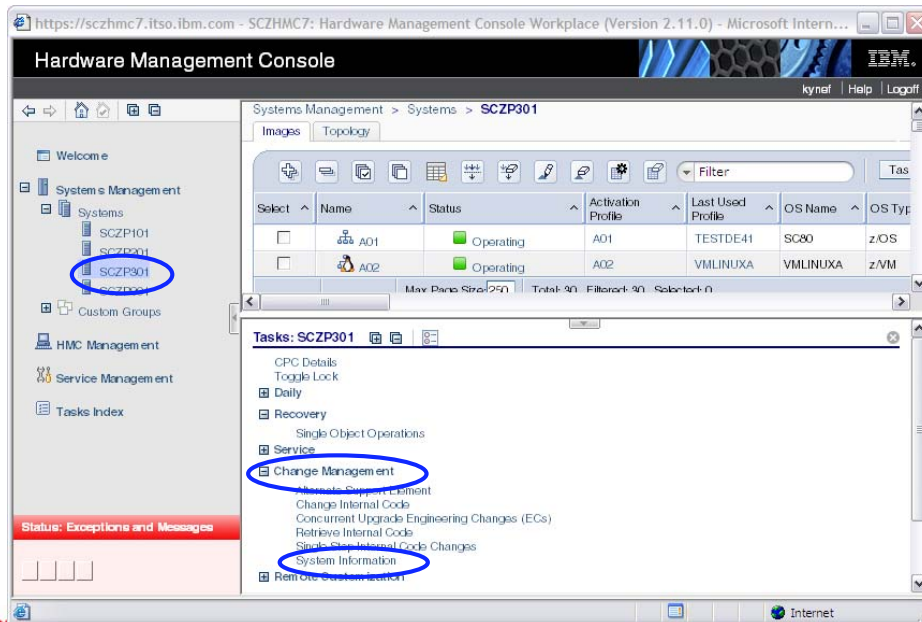
# CF Service Management

Finally, select the content you are interested in



# CF Service Management

And how do I find out MY current service level (in HW terms)?



## CF Service Management

And how do I find out MY current service level (in HW terms)?

The screenshot shows the 'System Information - SCZP301' panel in a Microsoft Internet Explorer browser. The 'Machine Information' section displays: EC number: N29802, LIC control level: 0003, Engineering Changes AROM Type: 2817, Model number: M32, Serial number: 0000200B3BD5, and Version: 2.11.0. Below this is a table of ECs with columns for Select, EC Number, Retrieved Level, Installable Concurrent, Activated Level, Accepted Level, and Description. The row for EC N29796 003 is highlighted with a blue oval, and its status is labeled 'ACCEPTed'. Other rows have labels 'RECEIVED' and 'APPLYed' above them.

Select	EC Number	Retrieved Level	Installable Concurrent	Activated Level	Accepted Level	Description
<input type="radio"/>	N29789					CHANNEL DIAGS
<input type="radio"/>	N29790 001	001	001	001	001	PCX LIC
<input type="radio"/>	N29791					OSA Express3 ICC
<input type="radio"/>	N29792 001	001	001	001		OSA Express3 Networking
<input type="radio"/>	N29793					OSA Express3 CDLC
<input type="radio"/>	N29794 006	006	006	006	003	FCS Ficon Express2 and Express4 LIC
<input type="radio"/>	N29795 006	006	006	006	000	FCS Ficon Express8 LIC
<input checked="" type="radio"/>	N29796 003	003	003	003	002	CFCC LIC
<input type="radio"/>	N29797 004	004	004	004	003	LPAR HV LIC
<input type="radio"/>	N29798					ESCON CHANNEL CODE LIC
<input type="radio"/>	N29799 035	035	035	035	026	I390/PU-ML LIC
<input type="radio"/>	N29800					SE LIC Alert

## CF Service Management

Latest news and recommendations:

- ALL z10 Customers should be on Driver 79. Previous Driver level is stabilized and all future fixes and enhancements will be to Driver 79.
- On Driver 79, CFCC should be on AT LEAST MCL N24403.006 (Service Level 2.25). 007 (Service Level 2.26) is the optimum level.
  - If you don't know the Driver level of your CPC, you can get it from the System Information panel using a Service HMC userid.
- Except in an exceptional situation, new CFCC MCLs can be applied and activated WITHOUT restarting the CF LPARs - this is known as "concurrent apply".

## Structure location and CF Service Levels

### A word to the wise.....

- It transpires that prior to CF Level 16, Service Level 2.25, CFCC provided incorrect info to XES that made all CFs look like internal ones. When all CFs look the same, structures will be allocated based on your PREFLISTs.
- Service Level 2.25 fixes this, so that XES now has accurate information (this is GOOD). However, if you have 2 external CFs, and you upgrade one to Service Level 2.25, XES will now think that he has 1 internal and 1 external CF. For some structures, XES will prefer an external CF, so this could result in your PREFLIST for those structures being overridden until both CFs are on 2.25.
- If this affects you, you can move structures back to where you want them using SETXCF START,REBUILD,STRNAME=xxxx,LOC=OTHER

## CF Service Management

### Summary:

- Make sure you are on latest Driver for your CPC.
- Monitor ResourceLink for new CFCC MCLs and see if any of the fixed problems are relevant to you.
- Always upgrade just one CF at a time.
- Try to avoid running with different CF Levels OR Service Levels for more than 4-6 weeks.
- Make sure that the hardware CE knows who to talk to if there is any service that affects the CFCC Service Stream. CEs are experts on hardware, but might have little knowledge of what CF exploiters you are using.



International Technical Support Organization

ibm.com/redbooks

## Implementing SMF logstream mode



© 2010 IBM Corporation. All rights reserved.

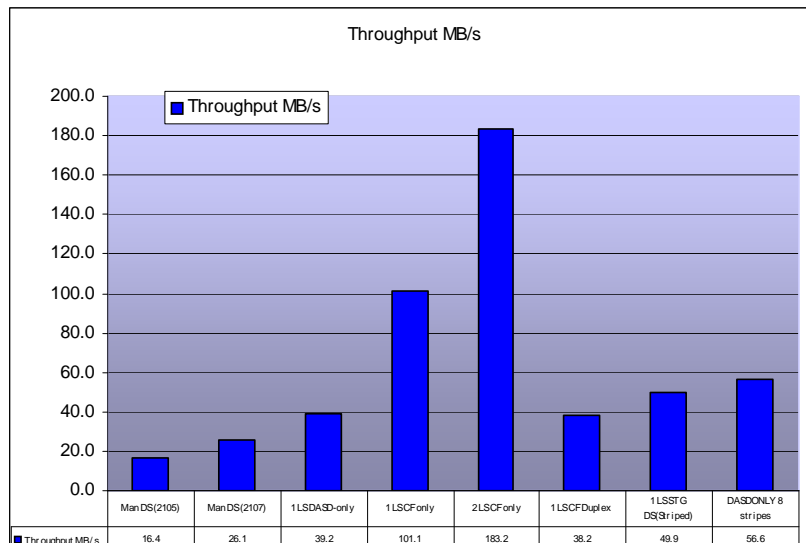
ibm.com/redbooks

International Technical Support Organization



## SMF logstream mode

### Why would I be interested in writing SMF data to log streams?



©2010 IBM Corporation. All rights reserved.

## Intro

To address the problem that some customers are creating SMF data faster than it can be saved to the SYS1.MAN data sets, z/OS 1.9 introduced the ability to write SMF records to log streams.

The basic function and the performance were good, but usability was equivalent to dataset mode. As a result, not many customers implemented the function at the time.

z/OS 1.11 delivered usability improvements (which are rolled back to 1.9), and z/OS 1.12 introduced yet more improvements (some of which are also rolled back to 1.9).

As a result, some form of migration from dataset mode to logstream mode is more now viable for many customers.

## Residency

The usability issues have now largely been addressed, however, the supporting documentation focuses on the creation of the SMF data, and does not address the bigger picture of the full lifecycle of your SMF data:

- This is how the SMF manual has always been positioned - information provided is similar to that provided for dataset mode.

To help customers determine if SMF logstream mode is right for them, and to help achieve an easier migration, the ITSO ran a residency in 2010 to create a Redbook on optimizing this new capability.

- This material is based on our experiences during that residency.
- Thanks to Dagmar Fischer, Jean-Marc Girona, Walter Klaey, Lennart Lundgren

## Dataset mode constraints

### Performance challenges of current SMF mechanisms:

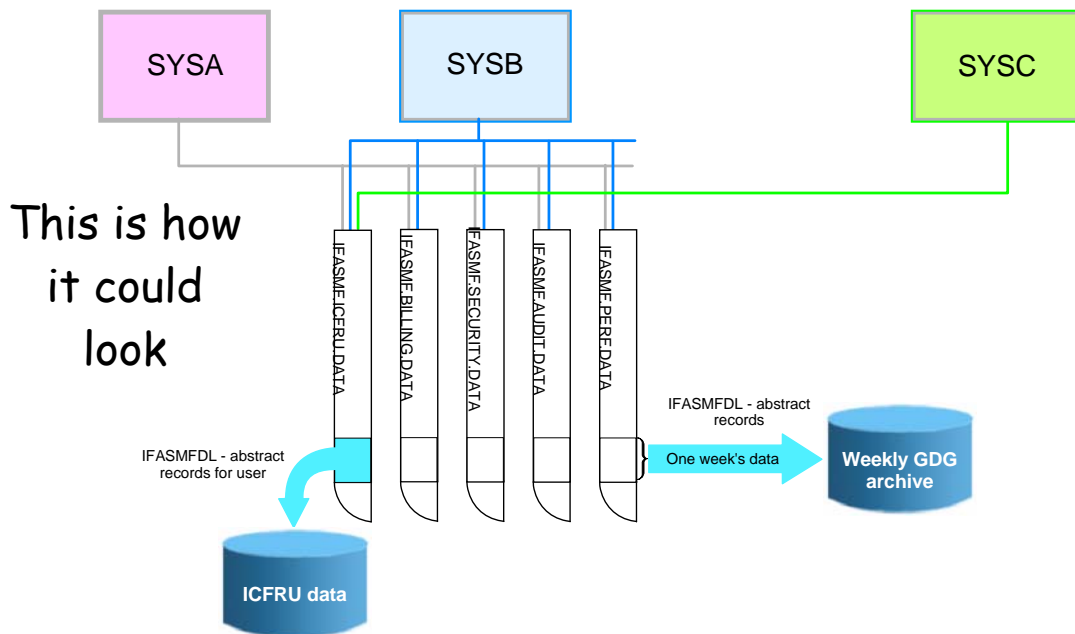
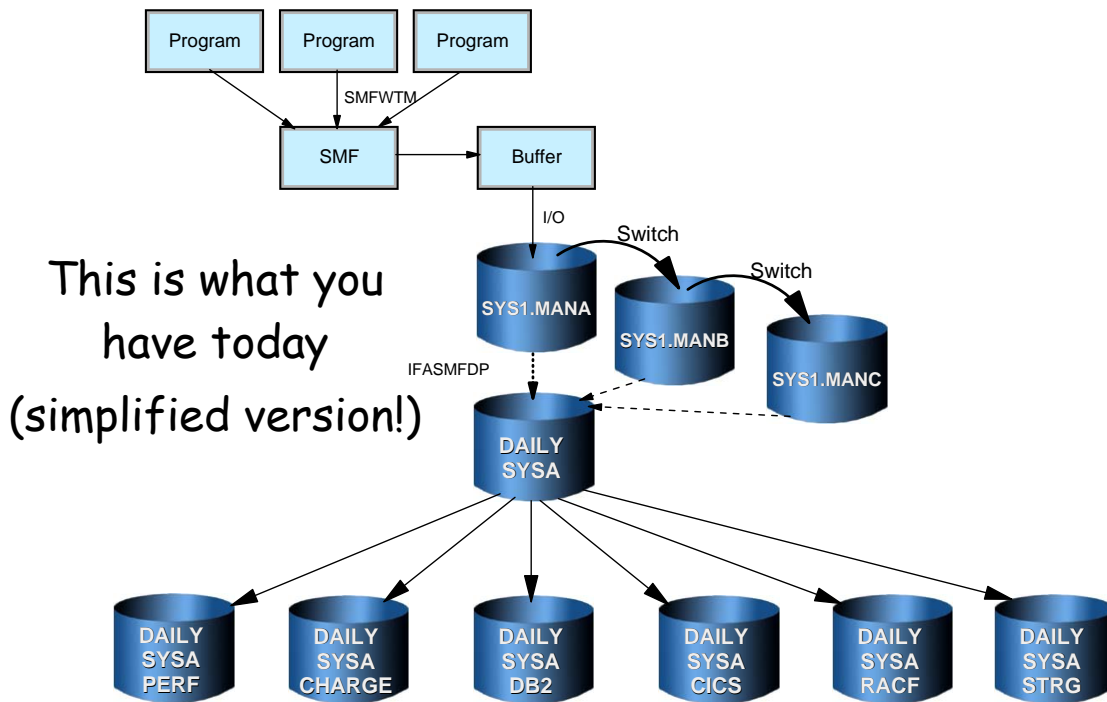
- The volume of SMF records that can be saved is gated by the performance of the DASD containing the SYS1.MAN data sets:
  - SMF doesn't support extended format data sets, so no striping support and no 4GB data set support
  - Only **one** active SYS1.MAN data set per system
  - If SMF data is consistently created faster than it can be saved to DASD, you will eventually start losing it (when the buffers fill)
    - Many installations have to give up collecting useful data in order to ensure vital SMF data isn't discarded
- The amount of buffer space is limited to 1GB
- There is only a single SRB, so only 1 I/O can be driven at a time
- SYS1.MAN data sets contain all record types for one system, but you probably want just some record types, but for the whole sysplex

## SMF logstream mode

### Advantages of logstream mode:

- Far better performance.
- Hugely scalable.
- SMF records are available for use much sooner.
- Records can be written directly into a sysplex-wide repository.
- FAR better support of no-buffer conditions - can have multiple NOBUFFS rules.
- Potential to provide better resiliency for critical record types.
- Management is conceptually much simpler, once you get past the complexity of having some data in GDGs and some in log streams.
- An ARCHIVE will fail if all records are not saved. An IFASMFDP ALL will happily delete records from SYS1.MAN Even if you did not save them.





## SMF logstream mode

### What are the options:

- Continue to use *SYS1.MAN* data sets (no change)
- Replace *SYS1.MAN* with 1 or more log streams and empty those into GDGs on a regular basis as you do with *SYS1.MAN* today
- Replace *SYS1.MAN* with log streams, keep some SMF record types in the log streams until they expire, and move the remainder to GDGs
- Replace *SYS1.MAN* with log streams and leave ALL the records there until they expire

Too much choice! Which one is right for me?

## SMF logstream mode

To arrive at the configuration that is best for *you*, you need to understand how your SMF data is being used today, AND you must understand how the characteristics of Logger interact with how you use your SMF data....



## SMF logstream mode

- A log stream contains all the data from the oldest not-yet-deleted record, right up to the just-written ones (sort of like a single data set that contains all generations of a GDG AND SYS1.MAN). So if you read an entire log stream, you may get much more data than if you read just one generation of a GDG.
- A log stream cannot have gaps. If you delete a log block, that block AND ALL PRECEDING ONES will be deleted.
- There is no concept of "switching", so you need to find some other trigger for SMF-related processing.
- Logger deals in terms of log blocks. It has no understanding of the contents of the log blocks. But SMF records are typically processed at the record level.
- DASDONLY log streams can only be accessed by one system at a time.

## SMF logstream mode

### What attributes make a particular record type a better candidate for keeping in GDGs?

- Records are used by many different programs, jobs, and users
  - JCL changes will be required to retrieve the data from log streams instead
- Users regularly process historical records and records are kept for a long time
  - Having some (older) data in GDGs and some data (more recent) in log streams is more complicated for users
- You don't have access to the JCL that processes these records (for example, daily/weekly/monthly jobs in TWS)
  - JCL changes *are* required, so you must be able to access the JCL
- The users are not familiar with JCL or are not IT people
  - Transitioning from GDG to log stream is easier if the users understand JCL

## SMF logstream mode

What attributes make a particular record type a better candidate for keeping in GDGs?

- Records are *accessed* from more than one sysplex, or from a sysplex other than where they were created
  - Log streams can only be accessed by one sysplex
- The current GDG contains data from multiple sysplexes
  - Log streams can only contain data from one sysplex
  - One example would be SCRT data for IBM - you would want to extract the Type 70 and 89 records to installation-wide sequential data sets
- GDG contains data from a GDPS/PPRC K system AND it is necessary to have all that data in one repository
  - GDPS/PPRC K systems don't run Logger so SMF records from the K system can't go in a log stream

## SMF logstream mode

What are the options if you want to process the data from the log stream but without changing your programs?

- Insert an IFASMF DL step to extract the records you want into a temporary sequential data set (we'll come back to this later)
- Change the existing JCL to point at the log stream instead of the existing data set (we'll come back to this as well):

```
//STEP1 EXEC PGM=IEBGENER
// * CHANGE LOGSTREAMNAME TO REQUESTED LOG STREAM NAME
//SYSUT1 DD DSN=IFASMF.STRIPE.TYPDFLT,DISP=SHR,
// DCB=(RECFM=VB,BLKSIZE=32760),
// SUBSYS=(LOGR,IFASEXIT,'FROM=(2010/265,19:10),TO=(2010/265,19:15)')
//SYSUT2 DD DSN=KYNEF.SMF.RECORDS,DISP=OLD
//SYSRINT DD SYSOUT=*
//SYSIN DD DUMMY
```

It's not a transparent replacement for IFASMFDP, but it IS possible...

## SMF logstream mode

We believe that most installations will have some SMF record types that are best suited to staying in GDGs, and some types that are best suited to staying in log streams.

If the benefits of logstream mode are attractive, you have two fundamental decisions to make:

- For each record type, is it best suited to long-term storage in a log stream or in a sequential data set?
- How many log streams you require and what are their attributes

## SMF logstream mode

How do I identify which category each record type falls into?

- The upcoming SMF Redbook (SG24-7919) provides tools that will help you identify:
  - Which data sets contain SMF records
  - The names of the jobs and programs that access those data sets
  - The RACF userids associated with those programs
- The tool is not comprehensive, but it should go a long way towards helping you understand who is using your SMF data

## SMF logstream mode

First pass identifies every run of IFASMFDP and the data sets created in the run (which *probably* contain SMF records):

### SMF dump program executions

Day	Time	SysID	Jobname	UserID	Program	I/O data set name	Type	Rfm	Gen	Retd
243	030000	ZT01	DUMPHY	SYSSTC	IFASMFDP	SYS1.ZT01.MAN2 SMF.DUMP.ZT00PLEX	MAN GDG VBS			250
243	065744	ZT01	LENNARTA	LENNART	IFASMFDP	SMF.DUMP.ZT00PLEX SYS2.R141562.F2	GDG VBS SEQ VBS			250
243	065931	ZT01	LENNARTA	LENNART	IFASMFDP	SMF.DUMP.ZT00PLEX SYS2.R141562.F2	GDG VBS SEQ VBS			250
243	065955	ZT01	LENNARTA	LENNART	IFASMFDP	SMF.DUMP.ZT00PLEX SYS2.R141562.F3	GDG VBS SEQ VBS			250
243	070018	ZT01	LENNARTA	LENNART	IFASMFDP	SMF.DUMP.ZT00PLEX SYS2.R141562.F4	GDG VBS SEQ VBS			250

## SMF logstream mode

Next pass identifies any other program that accesses the SYS1.MAN data sets directly

### Other jobs/programs accessing SMF MAN data sets

Day	Time	SysID	Jobname	UserID	Program	I/O data set name	Type	Rfm	Gen	Retd
190	151923	ZT01	SMF	SYSSTC		SYS1.ZT01.MAN2	MAN			
243	074239	ZT01	LENNARTQ	LENNART		SYS1.ZT01.MAN1	MAN			
					SMFDS1	LENNART.TEST.SMFDS1	SEQ VB			116

## SMF logstream mode

Final pass identifies the jobs, programs, and users that read the data sets that are created by the programs in either of the first two reports

Jobs using SMF data

Day	Time	SysID	Jobname	UserID	Program	I/O data set name	Type	Rfm	Gen	Retd
243	072846	ZT01	TDSSMFC1	LENNART	DRLPLC	SMF.DUMP.ZT00PLEX	GDG	VBS	250	
243	074239	ZT01	LENNARTQ	LENNART	IEBGENER	LENNART.TEST.SMFDS1	SEQ	VB		116
243	065222	ZT01	LENNART		IKJEFT01	LENNART.SMFCOPY	DTG	VB		2
243	080252	ZT01	LENNARTA	LENNART	IKJEFT01	SMF.DUMP.ZT00PLEX	GDG	VB	250	
243	080805	ZT01	LENNARTA	LENNART	IKJEFT01	SMF.DUMP.ZT00PLEX	GDG	VB	250	
243	081146	ZT01	LENNARTA	LENNART	IKJEFT01	SMF.DUMP.ZT00PLEX	GDG	VB	250	
243	081916	ZT01	LENNARTV	LENNART	VBSTOVV	SYS2.R141562.F4	SEQ	VBS		

## SMF logstream mode

We also provide an accompanying spreadsheet to help you document:

- For each record type, is it collected? If not, why not?
- What sequential data set(s) does it reside in (need to get this manually)
- What programs access each data set
- What jobnames are associated with those programs
- What userids are associated with the jobs
- How long is the data set retained for
- Does it contain records from 1 system or the whole sysplex?
- Is it acceptable to lose some of these records in case of a double failure?
- Is this a critical record for you?
- Which log stream this record type should go in

Once you have completed this spreadsheet, you are ready to move forward.

## SMF logstream mode

### How MANY log streams? Why have more than one log stream?

- Performance/scalability.
- Record types that will be immediately moved to GDGs should not be in the same log stream as record types that will stay in a log stream.
- Critical record types should go in a log stream that uses a staging data set for greater resiliency. Less critical ones can go in a CF-only log stream and get better performance.
- All the records in a given log stream should have similar retention requirements.
- Sensitive records, those with limited access, should go in a separate log stream.
- If you want system-specific log streams (but not sure why you would want that).
- Try to avoid very large Logger structures

## SMF logstream mode

### How do I handle the transition to logstream mode across my whole sysplex?

- Suggest to start by just replacing the SYS1.MAN data sets with your target log stream layout, and empty **all** the log streams into the existing sequential data sets.
- Switch each system over to log stream mode one-by-one until the whole sysplex is migrated.
- Then, for the record types that you plan to keep in log streams, simply stop running the ARCHIVE jobs and change any jobs that used to use the corresponding sequential data sets to use the log streams instead

## SMF logstream mode

Having identified which log streams you will have, and the mapping of record types to log streams, the next step is to size the log streams:

- The Redbook provides a tool to help you with this task as well...

There is one tool to turn an IFASMFDP report into CSV format. And a spreadsheet that lets you assign a log stream name to each record type and then provides you with information to help you size each log stream using CF Sizer.

The book also provides a tool that reads a data set containing SMF records and provides information about the average and peak creation rate (in bytes) of each record type.

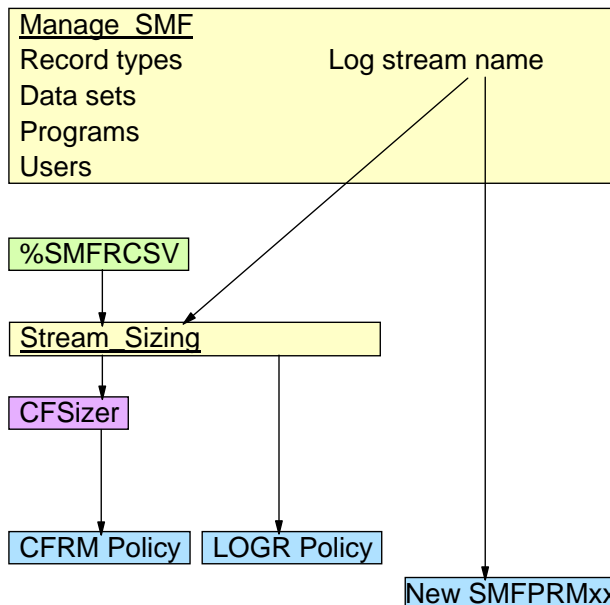
## SMF logstream mode

SMFPRMxx  
%SMFDSACC  
Study JCL

IFASMFDP Report

Relationship of ITSO tools to migration process

- Spreadsheet
- Rexx exec
- Web tool
- z/OS File





## SMF logstream mode

What type of log streams should I use?

- CF-only? CF with staging data set? DASDONLY?

Recommend that all log streams are CF log streams

- This allows you to access any log stream from any system in the sysplex

Recommend that log streams containing critical records be defined to use staging data sets (STG\_DUPLEX(YES), DUPLEXMODE(UNCOND)) - all other log streams should be CF-only.

Recommend that all log streams are sysplex-wide, unless you have a special reason for keeping them separate

## SMF logstream mode

You now have the information you need to define the log streams and associated structures, and the log stream definitions in the SMFPRMxx member.

What else is there to consider?



## SMF logstream mode

### Some implementation considerations....

#### - IFASMFDP has three options:

- **DUMP** Copy SMF records but don't delete them in input file
- **CLEAR** Delete SMF records in input file without copying them
- **ALL** Copy SMF records from the input file and then delete them

#### - IFASMFDL has three options:

- **DUMP** Copy SMF records from the named log stream(s)
- **DELETE** Delete log blocks from the beginning of the log stream to the specified end date and time
- **ARCHIVE** Copy log blocks from the beginning of the log stream to the specified end date and time then delete them.

- Note that DELETE and ARCHIVE work on a log block basis. Times and dates you provide are matched against the log block timestamp, NOT the SMF record timestamps.

## SMF logstream mode

### How much data is extracted from the log stream?

It depends!

They don't all START at the same place:

- ARCHIVE and DELETE ALWAYS start at the beginning of the log stream, ignoring any start date and time you specify
- DUMP starts at the specified start date and time, or at the beginning of the log stream if no start date and time are provided
- IFASEXIT starts at the beginning of the log stream, OR the date and time specified on the SUBSYS=LOGR DD statement

## SMF logstream mode

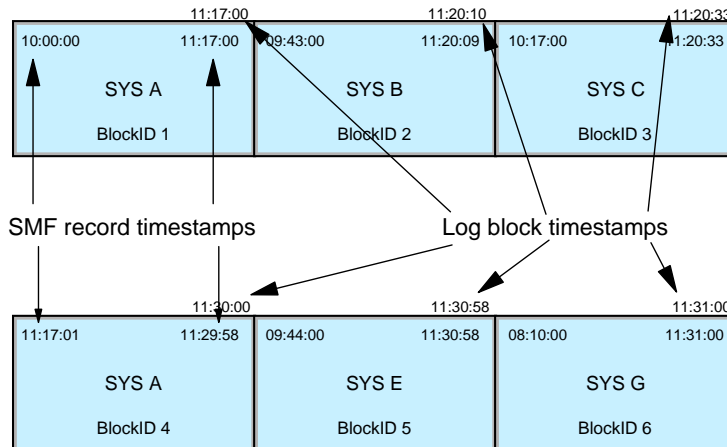
And they don't all end at the same place:

- ARCHIVE and DELETE currently ALWAYS read all the way to the end of the log stream:
  - The END date and time you specify controls what is deleted - NOT what is read.
- DUMP reads all the way to the end of the log stream, UNLESS you explicitly specify SMARTENDPOINT (it is NOT the default), in which case it reads two hours past the END date and time you specify
  - The SMARTENDPOINT value defaults to two hours, but APAR OA34374 will let you override that.
- IFASEXIT currently ALWAYS read all the way to the end of the log stream.

You may be wondering WHY they do this....

## SMF logstream mode

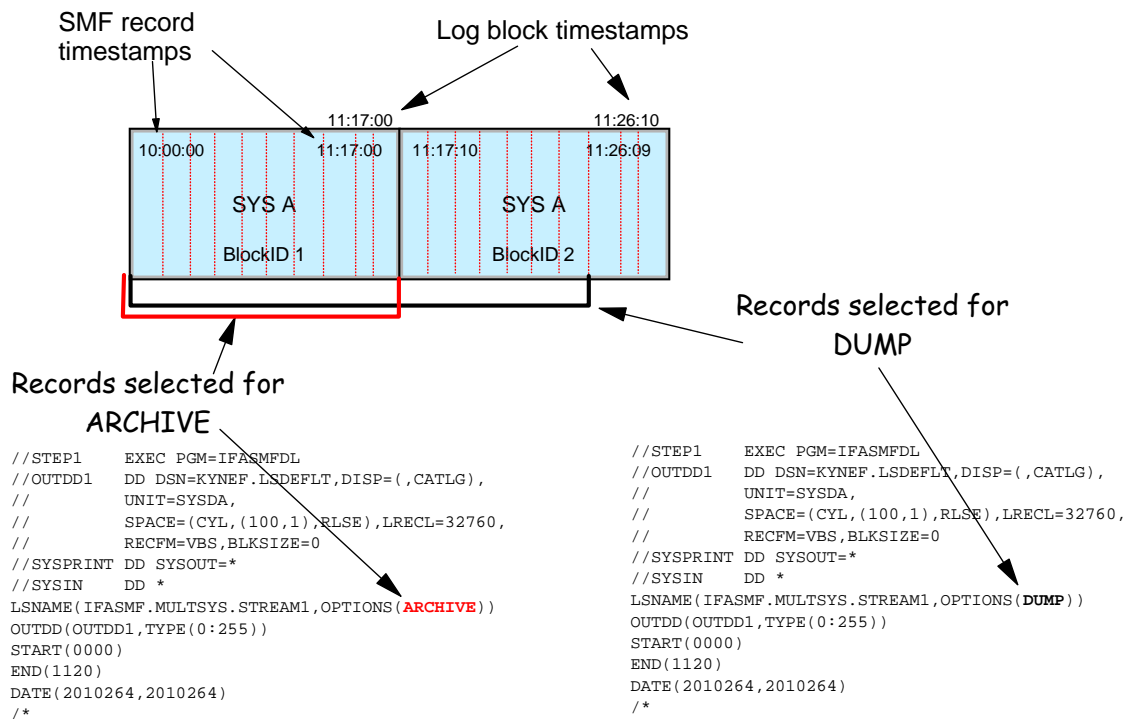
Consider these log block contents:



If you specify end time of 11:00, when should DUMP stop reading? How about ARCHIVE or DELETE?

## SMF logstream mode

Because ARCHIVE (and DELETE) work at the log block level (because you can't delete half a log block), running an ARCHIVE and a DUMP with the same parameters might result in a slightly different number of SMF records in the output file.



## SMF logstream mode

When you run an **IFASMF DL ARCHIVE**, there **MUST** be an **OUTDD** for every record in the specified time range:

- If **ARCHIVE** encounters any records that do not have a destination, the **ARCHIVE** will fail
- This is a **GOOD** thing, because it ensures that you can't accidentally delete **SMF** records. If there is a record in the log stream that you don't want to save, then you probably should not be collecting it in the first place....

## SMF logstream mode

When you run a **DELETE** or an **ARCHIVE**, the job will currently fail if the **DELETE** or **ARCHIVE** would delete **EVERY** log block in the log stream:

- Effectively this means that every **ARCHIVE** or **DELETE** must specify an **END** date and time
  - Specifying **RELATIVEDATE(BYDAY,0,1)** sets the **END** time to be the start time of the program. If the log stream is very active (or you specify a very small **MAXDORM**) this may allow you to **ARCHIVE** regularly without specifying an **END** time.
- There is an acknowledged marketing requirement (**MR0127106638**) - to make it acceptable to delete every log block from the log stream:
  - However, even if this is addressed, using **RELATIVEBYDATE(BYDAY,0,1)** is still a good idea, so the **IFASMF DL** doesn't run forever, constantly chasing the end of the log stream.

## SMF logstream mode

In order to minimize IFASMF DL job elapsed times, we recommend archiving data from the log stream (if that is what you are going to do) frequently - perhaps every hour:

- Remember that your SMF data sets are usually processed as soon as they switch - you don't wait until midnight to empty them all. If you wish to simply replace your SYS1.MAN with log streams, and keep all your GDG-based processes, then you should treat the log stream in a similar way to how you treat SYS1.MAN (frequent offloads)

## SMF logstream mode

For SMF records that you will leave in the log stream, use **RELATIVEDATE** to minimize JCL changes when extracting records from the log stream.

- For example: `RELATIVEDATE(BYWEEK,2,1)`
- **RELATIVEDATE** lets you specify:
  - What units you want to extract data in (in terms of days, weeks, or months)
  - How far back in the log stream processing should start
  - How far forward from there it should go.
  - Example above says to start processing at 00:00:01 of 2 Sunday's ago, and extract 168 hours worth of records
- Can be used to replace JCL like: `DSN=SMF.RMF.WEEKLY(-1)`

## SMF logstream mode

If you use IEFU29 exit, or data set switch messages to trigger IFASMFDP jobs, you will need to use an alternate (because there is no data set switch when using log streams):

- There is an IEFU29L exit that is invoked every time you do an I SMF
  - A sample is provided in the SMF Redbook

## SMF logstream mode

Because having SMF data in a log stream is a) a different paradigm than having it in data sets, and b) this exploits all new code in SMF, we had some unexpected experiences during the residency...

## SMF logstream mode

We were used to being able to run jobs against SMF GDGs from anywhere in the sysplex.

But if you have a DASDONLY log stream, that can only be used by one system. So put that system name in the log stream name to make it obvious that it is only for use by that system. For sysplex log streams, put sysplex name in the logstream name to help identify who owns offload data sets.

## SMF logstream mode

**IFA832I INVALID PARAMETER COMBINATION FOR ARCHIVE OPTION** is a VERY popular message - covers a plethora of problems.

- Usually they are related to breaking some SMF/Logger rule, for example:
  - Shouldn't specify a date range on the OUTDD statement when doing an ARCHIVE
  - Doing an ARCHIVE without RELATIVEDATE or DATE specified
  - Getting an abendB37 on the OUTDD data set
  - The granularity of these messages will be improved in the future

**If you specify a START date or time on an ARCHIVE, it will ignore them, but not tell you that it ignored them**



## SMF logstream mode

You can define both log streams and SMF data sets in your SMFPRMxx member.

The RECORDING parameter controls whether SMF writes to log streams or data sets.

You can switch recording modes using the SETSMF command.... HOWEVER, this requires that SMFPRMxx contains PROMPT(xxx).

If you say PROMPT in SMFPRMxx, you will get a prompt during IPL

- In z/OS 1.12 and later, you can use the Auto Reply function to reply to the SMF messages.

The other option is to maintain two SMFPRMxx members

## SMF logstream mode

Another lesson we learned is to have LOTS of DASD space available for the SMF Staging and Offload data sets:

- While you are testing, you will probably leave data in the log streams longer than you may in production. This will result in many offload data sets
- You will probably try different combinations of log streams. If you don't delete the old ones as you go along, they will be taking up disk space as well
- If you want to test striping, you will need several volumes in the storage group



## SMF logstream mode

When it comes time to do the cutover, we recommend making the switch at a time when no SMF jobs are scheduled - for example, just after midnight, when all the daily SMF jobs start is NOT a good time.

You should have a backout process that is documented and tested.

- Especially, remember to extract all the SMF records from whatever medium you just switched away from

## SMF logstream mode

Trying to understand what is going on....

You should do a lot of experiments in your sandbox sysplex, to get used to the new paradigm. One of the challenges as you do this is understanding what data is in the log streams, and what was extracted....

## SMF logstream mode

One thing you will want to see is what log streams are currently defined (because you will probably make several changes as you experiment):

All SMF log streams  
MUST start with  
IFASMF

```
D LOGGER,L,LSN=IFASMF.*
IXG601I  11.09.57  LOGGER DISPLAY 159
INVENTORY INFORMATION BY LOGSTREAM
LOGSTREAM          STRUCTURE          #CONN  STATUS
-----          -
IFASMF.STRIPE.TYPDFLT  *DASDONLY*          000001  IN USE
  SYSNAME: #@$2
  DUPLEXING: STAGING DATA SET
  GROUP: PRODUCTION

NUMBER OF LOGSTREAMS:  000001
```

## SMF logstream mode

And you will probably want to see the date range for the SMF records in each log stream, so you can test DUMP and ARCHIVE processes:

```
//KYNEFI JOB (0,0),CLASS=A,MSGCLASS=X,NOTIFY=&SYSUID
//STEP1 EXEC PGM=IXCMIAPU
//SYSPRINT DD SYSOUT=*
//SYSIN    DD *
  DATA TYPE(LOGR) REPORT(YES)
  LIST LOGSTREAM NAME(IFASMF.*) DETAIL(YES)
```

This results in a report like this:

## SMF logstream mode

```
LOGSTREAM NAME(IFASMF.STRIPE.TYPDFLT) STRUCTNAME() LS_DATACLAS(LOGR24KS)
LS_MGMTCLAS() LS_STORCLAS() HLQ(IXGLOGR) MODEL(NO) LS_SIZE(100000)
STG_MGMTCLAS() STG_STORCLAS() STG_DATACLAS(LOGR4KS) STG_SIZE(150000)
...
```

Log stream definition

LOG STREAM DATA SET INFO:

STAGING DATA SET NAMES: IXGLOGR.IFASMF.STRIPE.TYPDFLT.<SUFFIX>

Staging data set info

DATA SET NAMES:

IXGLOGR.IFASMF.STRIPE.TYPDFLT.##\$#PLEX

Offload data sets info

Ext.	<SEQ#>	Lowest Blockid / Highest Blockid	Highest GMT / Highest RBA	Highest Local / System Name	Status
00001	A0000000	0000000000000001 00000000186D4C80	09/07/10 17:10:43 186E3E5B	09/07/10 13:10:43 ##\$3	
	A0000001	00000000186E3E5C 0000000035B9155C	09/07/10 17:11:01 1D4BC8DC	09/07/10 13:11:01 ##\$3	

Date range in each data set

## SMF logstream mode

And then when you run DUMP and ARCHIVE jobs, you want to be able to see what data was in the input file or log stream, and what was in the output.

For these, we used a set of IFASMFDP/IFASMF DL exits that provide similar information to IFASMFDP report, only in more detail, and separately for the input and output files.

- The exits can also write this information to SMF records which you could then postprocess to get insight into SMF record processing

## Output from exit

DETAILED ACTIVITY REPORT FOR OUTPUT DDNAME: DUMP0UT										
RECORD	RECORD	SYSTEM	RECORDS	BYTES	RECORD	RECORD	START	START	END	END
TYPE	SUBTYP	ID	WRITTEN	WRITTEN	MINLEN	MAXLEN	DATE	TIME	DATE	TIME
2	N/A	#@\$2	1	18	18	18	10.269	13:50:43.17	10.269	13:50:43.17
3	N/A	#@\$2	1	18	18	18	10.269	13:53:53.34	10.269	13:53:53.34
4	N/A	#@\$3	22	4730	215	215	10.250	14:51:58.40	10.250	23:53:56.44
5	N/A	#@\$3	22	3206	145	149	10.250	14:51:58.40	10.250	23:53:56.44
14	N/A	#@\$3	143	58036	344	656	10.250	14:55:22.41	10.250	23:55:22.47
15	N/A	#@\$3	97	35188	344	372	10.250	14:55:22.41	10.250	23:55:22.45
17	N/A	#@\$3	6	600	100	100	10.250	17:18:22.53	10.250	23:07:59.08
20	N/A	#@\$3	45	4135	91	99	10.250	14:53:00.17	10.250	23:23:55.52
23	N/A	#@\$3	57	14706	258	258	10.250	14:33:22.20	10.250	23:53:22.20
30	1	#@\$3	45	18040	400	408	10.250	14:53:00.17	10.250	23:23:55.52
30	2	#@\$3	24149	34732165	1119	11055	10.250	14:30:00.00	10.250	23:59:25.16
30	3	#@\$3	35	52909	1119	3415	10.250	14:53:01.13	10.250	23:23:56.44

### SUMMARY ACTIVITY REPORT

START DATE-TIME	09/07/2010-14:30:00			END DATE-TIME	09/26/2010-13:53:40		
RECORD	RECORDS	PERCENT	AVG. RECORD	MIN. RECORD	MAX. RECORD	RECORDS	RECORDS
TYPE	READ	OF TOTAL	LENGTH	LENGTH	LENGTH	WRITTEN	WRITTEN
0	2	.00 %	64.00	64	64	0	0
2	0					1	1
3	0					1	1
4	153	.00 %	215.00	215	215	22	22
5	130	.00 %	147.02	145	158	22	22
8	2	.00 %	14,084.00	14,084	14,084	0	0
11	1,172	.01 %	28.00	28	28	0	0

## Standard IFASMFDP report

## SMF logstream mode

### Summary:

- Moving to SMF logstream mode IS a significant project, however it provides numerous benefits:
  - Better performance for SMF
  - Better protection for critical SMF record types
  - More flexibility to collect more SMF record types if that would be beneficial for you
  - Possibility for an easier to use and manage SMF infrastructure
  - The opportunity to review existing SMF processing for exposures and inefficiencies

## SMF and Logger

### Important service:

- OA27037 - Roll z/OS 1.11 SMF enhancements back to R9 and R10.
- OA31737 - Roll SMARTENDPOINT support back to R9 and R10
- OA34734 - Enhance control over SMARTENDPOINT processing
- All service for component ID 5752SC100 (SMF)

## Mean Time to Recovery Improvements



## Mean Time to Recovery

In the z/OS 1.10 timeframe, IBM announced a multi-year journey to reduce the time that applications are unavailable when a z/OS system needs to be IPLed.

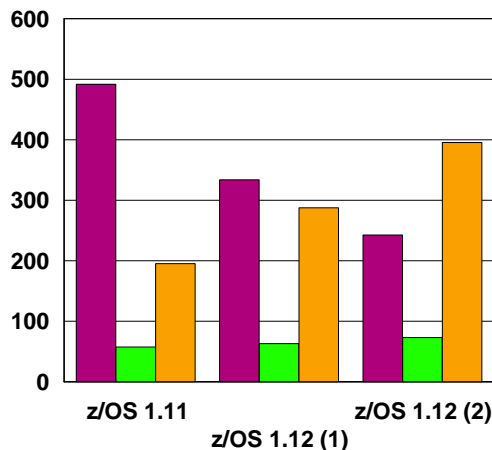
Enhancements were delivered in z/OS 1.10 and 1.11, and 1.12 includes further enhancements in the area of reduced bringup time for large DB2s, improved instrumentation, automatic WTOR replies, and greater parallelism during IPL.



## Mean Time to Recovery

Large DB2 startup time improvements - one set of measurements:

Large DB2 Startup Benefits in z/OS 1.12



■ Elapsed Time (secs)  
 ■ CPU Busy %  
 ■ ETR

Measurement taken using 96K non-SMS DB2 data sets

## Mean Time to Recovery

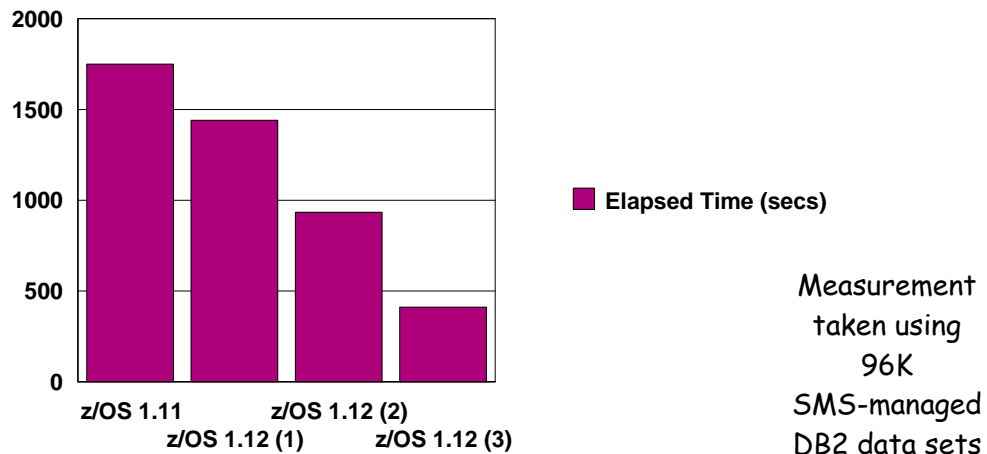
### What did they do to get these results?

- All measurements were done using:
  - USEZOSV1R9RULES(NO) in DIAGxx. The enhancements in this area in z/OS 1.12 REQUIRE the use of the new GETMAIN/FREEMAIN rules.
  - Enhanced Catalog Sharing was enabled for all measurements.
  - DB2 V9 with APARs PM00068, PM17542, and PM18557 (included in DB2 V10)
- The first z/OS 1.12 measurement (1) also had:
  - GRS APAR OA33633 (selectively lets SHARE ENQ requests skip ahead of pending EXCLUSIVE requests)
- The second z/OS 1.12 measurement (2) had the above changes PLUS MEMDSENQMGMT=ENABLE in ALLOCxx
  - MEMDSENQMGMT allows much more efficient mgmt of large numbers of data set requests

## Mean Time to Recovery

### Large DB2 startup time improvements - another set of measurements:

Large DB2 Startup Benefits in z/OS 1.12





## Mean Time to Recovery

### What did they do to get these results?

- All measurements were done using:
  - A different hardware environment
  - USEZOSV1R9RULES(NO) in DIAGxx.
  - Enhanced Catalog Sharing was enabled for all measurements.
  - DB2 V9 with APAR PM00068
  - SMS-managed DB2 data sets
- First measurement was z/OS 1.11 (base measurement)
- The first z/OS 1.12 measurement (1) had the same options.
- The second z/OS 1.12 measurement (2) had:
  - MEMDSENQMGMT=ENABLE and DB2 APAR PM17542
- The third z/OS 1.12 measurement (3) had:
  - GRS APAR OA33633
  - DB2 APAR PM18557

## Mean Time to Recovery

In addition to the Allocation-based enhancements in z/OS 1.12, there are also enhancements in the area of reducing the volume of data being written to SMF Type 30 records.

Even if you turn off Type 30 records completely, Allocation still needs to collect and maintain all the information that would go in the Type 30, in case you turn them on again in the future, so disabling Type 30 records does not result in significant savings..

Normally, Allocation will collect information to populate the EXCP section of the Type 30 record:

- You get one EXCP section per DD name/Device pair
- For SMS-managed data sets, by default you get one EXCP section per candidate device in the storage group
- Each EXCP section is 30 bytes long, but does NOT include the DSN

## Mean Time to Recovery

Dynamic Allocation (SVC 99) has been enhanced to allow the suppression of the EXCP sections (under the control of the caller of SVC99) on a DD level.

- AND, Allocation is aware that the EXCP sections of the SMF records will not be populated, so it does NOT collect that information either

This should result in savings in Allocation CPU, smaller volumes of SMF data, and reduced time to build the Type 30 records.

## Mean Time to Recovery

The focus of these enhancements has been DB2 because we are seeing ever-larger DB2 subsystems

However, all the enhancements use published interfaces, so other products can use them as well if they wish

## Mean Time to Recovery

z/OS includes a control block called the IPST that contains information about the elapsed time of various steps in the IPL process.

Information in the control block can be formatted using an IPCS command: VERBX BLSAIPST MAIN

## Mean Time to Recovery

This provides a report like:

```

IPLST000I z/OS          01.11.00 #@$2      20970005DE50    2 CPs
IPLST001I IPL started at: 2010/08/13 17:00:16.052
IPLST100I **** IPL Statistics ****
IPLST101I IEAIPL10      0.000   ISNIRIM - Read SCPINFO
IPLST101I IEAIPL20      0.000   Test Block storage to 2G
IPLST101I IEAIPL11      0.006   Fast FIND service
IPLST101I IEAIPL31      0.001   LOAD service
IPLST101I IEAIPL30      0.000   IPLWTO service
IPLST101I IEAIPL46      0.094   Read SCHIBs into IPL workspace
IPLST101I IEAIPL49      0.000   Process Load and Default parameters
IPLST101I IEAIPL50      0.021   IPL parmlib - process LOADxx and NUCLSTxx
IPLST101I IEAIPL51      0.000   System architecture
IPLST101I IEAIPL43      0.008   Find and Open IODF data set
IPLST101I IEAIPL60      0.001   Read NCRs from IODF
IPLST101I IEAIPL70      0.044   UIM environment - load CBD and IOS services
IPLST101I IEAIPL71      0.042   Build DFT for each device
  
```

The report is described in *System z Mean Time to Recovery Best Practices, SG24-7816*

## Mean Time to Recovery

In an attempt to make more information available to both developers and customers, z/OS 1.12 provides a new interface that allows programs to provide more information.

New interface is IEATEDS (macro is in SYS1.MACLIB). Allows much more information to be saved, for example:

- Component name
- Whether the event represents the start, a midpoint, or end of a process
- A load of user data
- Full timestamp
- SRB and TCB time

## Mean Time to Recovery

z/OS 1.12 also provides a compiled Rexx program that will extract the IEATEDS information from storage or a dump and create a data set containing:

- A human-readable report about the contents
- The same information in CSV format, for downloading to a spreadsheet

## Mean Time to Recovery

To run the program, allocate a VB 512 data set, and then issue:

-IEAVFTED DATASET('HLQ.DATASET.NAME')

- The resulting report (and CSV) combines information from the traditional IPST control block, and information created using the IEATEDS service.

- Unfortunately, the program requires the Full Rexx Runtime library (5695-014):

- If you try to run with the Alternate Library (the free one), you get:
  - EAGALT0300E Error 3 running compiled IEAVFTED, line 0: Program is unreadable
  - EAGALT0304I The program cannot run with the Alternate Library
- However, if you have one system that has the full library, you can take a dump and IEAVFTED will extract the information from the dump.

The resulting report looks like this:

## Mean Time to Recovery

```

*****
*
* IBM z/OS Timed Event Data Report
* Level: HBB7770-VL.03      Report Date/Time: 21 Sep 2010 12:55:54      Component Filter: ALL
* Sysplex: #@$#PLEX      System: #@$3      FMID: HBB7770 z/OS V01R12M00
* Machine: 2817-00223BD5 Online Standard CPs: 2 zAAPs: 2 zIIPs: 2
* IPL Start Date/Time: 21 Sep 2010 12:51:29.933539
*
*****

*****
*
* Total Timed Event Data Table Storage: 0006FE10
*
*****
*
* Timed Event Data Table - Component: XCF/XES      Address: 000001EF80301000
* Table Size: 00007F90      Register Date/Time: 21 Sep 2010 12:51:49.194001
* Requested MaxEvents:      185      Resultant MaxEvents: 185      NumEvents: Current: 76      Overflow: 0
*
*****

```

```

EntryNum:      1      Event Type/Thread: Start/0000000000FFF001/*.....0.*      Event Date/Time: 21 Sep 2010 12:51:49.194015
Description: Start of XCF/XES Initialization
HASN: 0001      PASN: 0001      Jobname: *MASTER*      TCB: 00FDB3B8      Module/Level/Offset: IXCI2RIM/HBB7770 /00000310
SRB/Task Time: 000000007BD384D4/00000002BA278288      User Data: 00000000 00000000 00000000 00000000 *.....*
OUXBFCON: 00:00:00.017152      OUXBFDIS: 00:00:00.046976      OUXBFMNO: 00:00:00.028000      OUXBFWAIT: 00:00:00.003200
Deltas:      IPL Start: 0 Days 00:00:19.260475      T.E.D. Registration: 0 Days 00:00:00.000014
      Thread Start Event: 0 Days 00:00:00.000000      Thread Prior Event: 0 Days 00:00:00.000000

```



# Mean Time to Recovery

You can also download the CSV part of the report into a spreadsheet and do a little color coding and sorting....

	B	C	D	E	F	G	H	I	J	K
1	Event Time	Date	Event Thread	Thread EBCDIC	Type	Description	Component	IPL Start Delta	Thread Start Event Delta	T.E.I. Registr Delta
116	14-02-50.199193	15-Mar-10	IEAVNP25	IEAVNP25	End	Process SVC=	IPST	57.835346	0.00904	
117	14-02-50.199193	15-Mar-10	IEAVNP05	IEAVNP05	Start	LPA, APF	IPST	57.835346	0	
118	14-02-50.369684	15-Mar-10	IEAVNPC5	IEAVNPC5	Start	Build Pageable Link Pack Area	IPST	58.005838	0	
119	14-02-51.484682	15-Mar-10	Reading Direct	Reading Directorie	Start	Build Pageable Link Pack Area	IPST	59.120835	0	
120	14-02-51.999386	15-Mar-10	Reading Direct	Reading Directorie	End	Build Pageable Link Pack Area	IPST	59.63654	0.514704	
121	14-02-51.999481	15-Mar-10	Validity check	Validity check Ali	Start	Build Pageable Link Pack Area	IPST	59.635634	0	
122	14-02-51.999516	15-Mar-10	Validity check	Validity check Ali	End	Build Pageable Link Pack Area	IPST	59.635669	0.000034	
123	14-02-52.011589	15-Mar-10	Non Packlist P	Non Packlist Proc	Start	Build Pageable Link Pack Area	IPST	59.647742	0	
124	14-02-52.012097	15-Mar-10	Non Packlist P	Non Packlist Proc	End	Build Pageable Link Pack Area	IPST	59.64825	0.000508	
125	14-02-52.012097	15-Mar-10	Load modules t	Load modules to F	Start	Build Pageable Link Pack Area	IPST	59.64825	0	
126	14-02-52.801429	15-Mar-10	Load modules t	Load modules to F	Mid	ASM Page-Out Wait Time	IPST	60.437582	0.789331	
127	14-03-08.557938	15-Mar-10	Load modules t	Load modules to F	End	Build Pageable Link Pack Area	IPST	76.194092	16.545841	
128	14-03-08.919958	15-Mar-10	IEAVNPC5	IEAVNPC5	End	Build Pageable Link Pack Area	IPST	76.556112	18.550274	
129	14-03-09.132255	15-Mar-10	IEAVNP05	IEAVNP05	End	LPA, APF	IPST	76.768408	18.933062	

	B	C	D	E	F	G	K	L	M	N
1	Event Time	Date	Event Thread	Thread EBCDIC	Type	Description	T.E.D. Registration Delta	Thread Prior Event Delta	Jobname	HASN
116	14-02-50.199193	15-Mar-10	IEAVNP25	IEAVNP25	End	Process SVC=		0.00904	*MASTER*	*0001*
117	14-02-50.199193	15-Mar-10	IEAVNP05	IEAVNP05	Start	LPA, APF		0	*MASTER*	*0001*
118	14-02-50.369684	15-Mar-10	IEAVNPC5	IEAVNPC5	Start	Build Pageable Link Pack Area		0	*MASTER*	*0001*
119	14-02-51.484682	15-Mar-10	Reading Direct	Reading Directorie	Start	Build Pageable Link Pack Area		0	*MASTER*	*0001*
120	14-02-51.999386	15-Mar-10	Reading Direct	Reading Directorie	End	Build Pageable Link Pack Area		0.514704	*MASTER*	*0001*
121	14-02-51.999481	15-Mar-10	Validity check	Validity check Ali	Start	Build Pageable Link Pack Area		0	*MASTER*	*0001*
122	14-02-51.999516	15-Mar-10	Validity check	Validity check Ali	End	Build Pageable Link Pack Area		0.000034	*MASTER*	*0001*
123	14-02-52.011589	15-Mar-10	Non Packlist P	Non Packlist Proc	Start	Build Pageable Link Pack Area		0	*MASTER*	*0001*
124	14-02-52.012097	15-Mar-10	Non Packlist P	Non Packlist Proc	End	Build Pageable Link Pack Area		0.000508	*MASTER*	*0001*
125	14-02-52.012097	15-Mar-10	Load modules t	Load modules to F	Start	Build Pageable Link Pack Area		0	*MASTER*	*0001*
126	14-02-52.801429	15-Mar-10	Load modules t	Load modules to F	Mid	ASM Page-Out Wait Time		0.789331	*MASTER*	*0001*
127	14-03-08.557938	15-Mar-10	Load modules t	Load modules to F	End	Build Pageable Link Pack Area		15.756503	*MASTER*	*0001*
128	14-03-08.919958	15-Mar-10	IEAVNPC5	IEAVNPC5	End	Build Pageable Link Pack Area		18.550274	*MASTER*	*0001*
129	14-03-09.132255	15-Mar-10	IEAVNP05	IEAVNP05	End	LPA, APF		18.933062	*MASTER*	*0001*

## Mean Time to Recovery

**IEATEDS Service is documented in *MVS Programming: Authorized Assembler Services Reference, Volume 2 (EDT-IXG)*, SA22-7610**

- Information about using the IEAVFTED exec is contained in the PROLOG for the IEATEDS macro in SYS1.MACLIB

**The information in the report helps you identify parts of the IPL process that have significant elapsed times.**

- For those parts, you may be able to see how much of that time is using CPU (if the program used the IEATEDS service).
- It may also highlight points where the IPL process is waiting for manual intervention...
  - Which brings us to....

## Mean Time to Recovery

**The Auto Reply facility in z/OS 1.12**

**One of the common causes of elongated IPLs is delays waiting for the operator to respond to a WTOR**

- Especially common example is if PROMPT is specified in the SMFPRMxx member

**One of the challenges is that automation has not yet been started, so is not able to help with prompts this early in the IPL process.**

- But the Auto Reply facility is available very early in the IPL process, so can provide fixed responses to common, expected prompts
  - Earliest point is just before XCF comes up. You can reply to the XCF Initialize or Join message but nothing before that. Watch for message:
  - CNZ2600I AUTO-REPLY POLICY ACTIVATED.



## Mean Time to Recovery

For prompts that always get the same answer, you can use an Auto Reply policy. For example:

```
AUTOR00 member
/*****/
/*                                     */
/*  AUTOREPLY OPTION                   */
/*                                     */
/*  Change activity:                   */
/*  08/12/10 Walter Klaey Initial setup */
/*****/
MSGID(IEE357A) REPLY('U') DELAY(1S)
MSGID(IEE956A) REPLY('U') DELAY(1S)
```

For more information on Auto Reply, see Paul's presentation from yesterday and the z/OS Init and Tuning Reference.

## Mean Time To Recovery

When it is initializing subsystems from the IEFSSN member, z/OS works as follows:

- Read line from IEFSSNxx member ←
- Add subsystem to list of defined subsystems
- If subsystem definition includes an initialization routine (INITRTN), start initializing subsystem
- Wait for subsystem initialization to complete

## Mean Time to Recovery

**z/OS 1.12 adds the ability to specify that subsystems should be initialized in parallel:**

- Only works with IEFSSN that uses the "new" keyword format.
  - Old positional format does not support parallel initialization
- Subsystems that do not use initialization routines are unaffected (they take very little time to process anyway)

**New keyword BEGINPARALLEL indicates at which point you want to start parallel initialization:**

- SMS must be specified before this keyword
- Any subsystem that has dependencies on another subsystem should be specified before BEGINPARALLEL
  - See Init and Tuning Reference for a full list of restrictions

## Mean Time to Recovery

**These enhancements are only the ones that are visible.**

**There is ongoing work to reduce elapsed time for all aspects of IPL and shutdown processing**

- Many have no externals and are not immediately obvious

**Future releases of z/OS and the major subsystems will deliver yet more enhancements in this area.**

## z/OS 1.12 PDSE enhancements

When a corrupt PDSE is detected in the link list during IPL, the system enters a wait state. In z/OS V1.12, the system issues a message identifying the corrupt PDSE prior to entering the wait state. This allows the user to attempt to restore the corrupt PDSE and re-IPL the system and avoid taking a stand alone dump to debug the problem.

## Predictive Failure Analysis

### Contents:

- Why IBM created Predictive Failure Analysis (PFA)
- A brief overview of the PFA functionality and infrastructure
- Which checks were already available in previous releases
- The enhancements in z/OS V1R12

## Software detected failures

### Masked failures

- Are detected and corrected by the software

### Hard failures

- The software fails completely, for example: a system kills a process. While this is unpleasant, it is quick and unambiguous

### Soft failures

- These are caused by abnormal, unexpected, or unusual (although valid) behavior and are hard to detect
- They can cause "sympathy sickness" in other parts of the system or sysplex and escalate to a point where the service is impacted
- They can be misleading - Product B fails trying to obtain common storage. But the reason for the failure is that Product A is using an unusually large amount of common
- It can be hard to determine which actions to take to recover from them

## Background

Review of a large number of incidents uncovered the following generic causes, that could all be categorized as soft failures:

- Damaged systems
  - Recurring and recursive errors caused by a defect in software components
- Serialization problems
  - Priority inversion
  - Classic deadlocks
  - Owner is gone but serialization not released
- Resource exhaustion
  - Physical and software resources
- Intermediate or unexpected states

Of the incidents in the study, these soft failures were responsible for 20% of problems, but 80% of business impact

## Predictive Failure Analysis (PFA)

The objective of PFA is to help you identify and eliminate soft failures that could eventually impact the proper functioning of the system if no action is taken

PFA tries to predict if a soft failure will occur in the future, and identify the cause:

- PFA uses:
  - Historical data
  - Machine learning and mathematical modeling
- To detect abnormal behavior and the potential cause
- The final PFA objective is to:
  - Convert a "sick but not dead" problem into a correctable incident

Introduced as an SPE in z/OS V1R10 and enhanced in V1R11 and again in V1R12

## The PFA Infrastructure - part 1

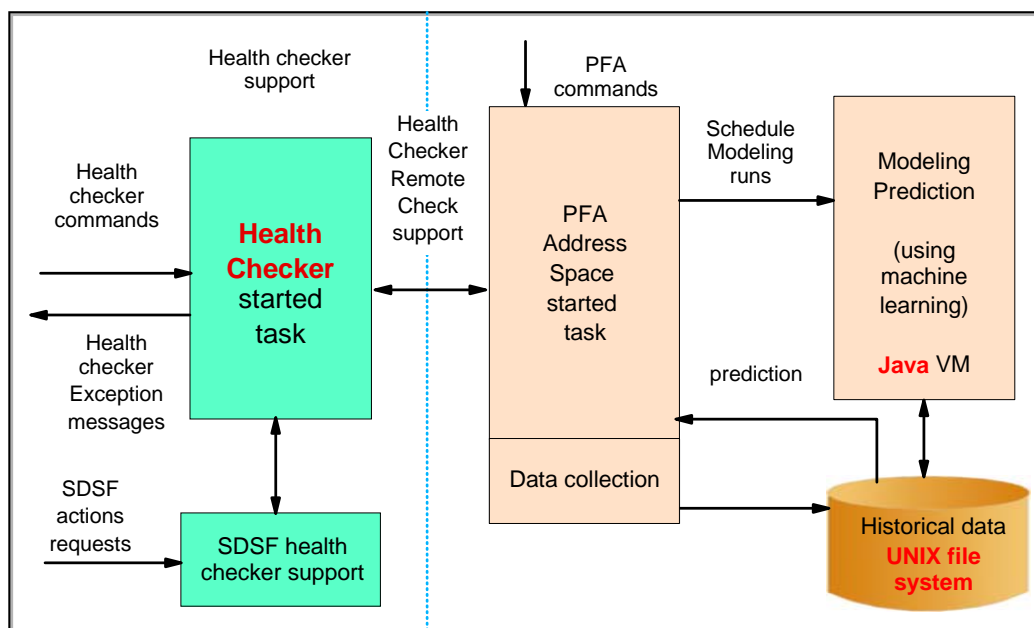
The PFA checks uses the Health Checker infrastructure (they run as remote checks):

- Therefore the Health Checker commands, report capabilities, and SDSF interfaces can be used

The PFA infrastructure consists of:

- The PFA address space:
  - Which connects to the Health Checker, displays the status of PFA, and launches the JVM that is used for modeling
- Check-specific code (these reside in the PFA address space):
  - To collect the data, model the data and create the predictions, and compare the actual situation to the prediction and issue messages and reports
- Historical data and predictions, stored in the UNIX file system, for use by the code that does the actual compare

## The PFA infrastructure - part 2



## The three parts of the PFA process

### ***Gathering*** data that will be used by the modeling and comparison processes:

- The data collection is specific for a certain check. For example, one check is responsible for monitoring common storage usage, and it gets its information from control blocks. A different check is responsible for monitoring message rates, and it uses a different mechanism for gathering its information.
- The gathering process is triggered by PFA every x minutes (the value for "x" is controlled by the PFA COLLECTINT parameter for each check):
  - Many of the checks use exits (such as IEFU82/83/84) to collect the requirement information on an ongoing basis. The COLLECTINT determines how frequently that information is gathered and input to the PFA database.

## The three parts of the PFA process

### ***Modeling*** the data to create a prediction:

- Each check has a modelling component that generates predictions of how that aspect of the system will look during the next modeling interval. By default, the prediction is updated every 6 hours (every 12 hours in z/OS 1.12). This can be changed using the PFA MODELINT parameter for each check.
  - In addition, if a compare indicates an exception, a new modelling run will be done and a new projection created, and the current situation compared to the new projection before an exception is raised.
  - PFA tries to select appropriate data for the modeling - for example, if it is invoked at 08:00, it will look at how the system looked yesterday at 08:00 and a week ago at 08:00.
- Each modelling invocation takes a few seconds



## The three parts of the PFA process

### *Comparing the actual to the predicted configuration:*

- The checks compare the current system data with the prediction and issue messages and reports when deviations are found.
- This is triggered by the Health Checker INTERVAL parameter for each check, and is initiated by the Health Checker. Some checks are run at the end of each data collection.

## The three parts of the PFA process

### *Controlling timing of events*

- Each subject area that PFA monitors has its own check. The initiation of the Collection and Modeling activities for each check are controlled by PFA. However, even though these are PFA parameters, the only way to SET those keywords is via a HealthChecker command:
  - F HZSPROC,UPDATE,CHECK(IBM PFA,PFA\_COMMON\_STORAGE\_USAGE),PARM('collectint(2)')
- Note that overrides to the PFA parameters only survive for the life of that instance of PFA - if you restart PFA, it reverts back to its default values.

## The three parts of the PFA process

You can display the settings of the **COLLECTINT** and **MODELINT** parameters with the **MODIFY PFA** command:

```
F PFA,DISPLAY,CHECKS,DETAIL
AIR018I 23:35:46 PFA CHECK DETAIL 998
CHECK NAME: PFA_COMMON_STORAGE_USAGE
  ACTIVE                : YES
  TOTAL COLLECTION COUNT : 78
  SUCCESSFUL COLLECTION COUNT : 78
  LAST COLLECTION TIME   : 09/20/2010 23:34:49
  LAST SUCCESSFUL COLLECTION TIME: 09/20/2010 23:34:49
  NEXT COLLECTION TIME   : 09/20/2010 23:36:49
  TOTAL MODEL COUNT      : 7
  SUCCESSFUL MODEL COUNT : 7
  LAST MODEL TIME        : 09/20/2010 23:27:48
  LAST SUCCESSFUL MODEL TIME : 09/20/2010 23:27:48
  NEXT MODEL TIME        : 09/20/2010 23:37:48
CHECK SPECIFIC PARAMETERS:
  COLLECTINT            : 2
  MODELINT              : 10
  COLLECTINACTIVE       : 1=ON
  DEBUG                 : 0=OFF
  THRESHOLD             : 2
```

## The three parts of the PFA process

Can also see the values using the **SDSF CK** command:

```
CHECK(IBMPPFA,PFA_COMMON_STORAGE_USAGE)
START TIME: 09/20/2010 23:35:48.338119
CHECK DATE: 20071101 CHECK SEVERITY: MEDIUM
CHECK PARM: modelint(10)
```

### Common Storage Usage Prediction Report

```
Last successful model time   : 09/20/2010 23:27:48
Next model time              : 09/20/2010 23:37:48
Model interval            : 10
Last successful collection time: 09/20/2010 23:34:49
Next collection time         : 09/20/2010 23:36:49
Collection interval       : 2
```

# The three parts of the PFA process

## Controlling timing of events

- The frequency at which the *comparison* to the predicted configuration for each check is run IS controlled directly by HealthChecker, using its **INTERVAL** keyword
  - *Additionally*, for most checks, a compare is run automatically at the end of each collection.
  - Recommend that you do NOT change the INTERVAL value in Health Checker.

# The three parts of the PFA process

The INTERVAL value can be shown and changed in the CK panel:

```

SDSF HEALTH CHECKER DISPLAY #@$# LINE 55-90 (158)
COMMAND INPUT ==>
NP
NAME Start-Date-Time Interval NextSch-Date
PFA_COMMON_STORAGE_USAGE 09/20/2010 23:43:56 0:01 09/20/2010 23
PFA_FRAMES_AND_SLOTS_USAGE 09/20/2010 23:43:43 0:01 09/20/2010 23
PFA_LOGREC_ARRIVAL_RATE 09/20/2010 23:32:41 0:15 09/20/2010 23
PFA_MESSAGE_ARRIVAL_RATE 09/20/2010 20:17:42 ONETIME ***** N/A **
PFA_SMF_ARRIVAL_RATE 09/20/2010 20:17:42 ONETIME ***** N/A **
RACF_FACILITY_ACTIVE 09/20/2010 18:17:34 24:00 09/21/2010 08
RACF_GRS_RNL 09/20/2010 18:17:34 8:00 09/21/2010 00
RACF_IBMUSER_REVOKED 09/20/2010 18:17:34 24:00 09/21/2010 08
RACF_ICHAUTAB_NONLPA 09/20/2010 18:17:34 24:00 09/21/2010 08
RACF_OPERCMDS_ACTIVE 09/20/2010 18:17:34 24:00 09/21/2010 08
RACF_SENSITIVE_RESOURCES 09/20/2010 20:37:51 4:00 09/21/2010 00
RACF_TAPEVOL_ACTIVE 09/20/2010 18:17:34 24:00 09/21/2010 08
RACF_TEMPDSN_ACTIVE 09/20/2010 18:17:34 24:00 09/21/2010 08
RACF_TS_AUTH_ACTIVE 09/20/2010 18:17:34 24:00 09/21/2010 08
RACF_UNIXPRIV_ACTIVE 09/20/2010 18:17:34 24:00 09/21/2010 08
RCF_PCCA_ABOVE_15M 09/20/2010 18:17:34 ONETIME ***** N/A **
RRS_ARCHIVECPSTRUCTURE 09/20/2010 18:17:34 8:00 09/21/2010 00
RRS_DUROFFLOADSIZE 09/20/2010 18:17:34 8:00 09/21/2010 00
RRS_MUROFFLOADSIZE 09/20/2010 18:17:34 8:00 09/21/2010 00
RRS_RMDATLOADDUPLXMODE 09/20/2010 18:17:34 8:00 09/21/2010 00
RRS_RMODOFFLOADSIZE 09/20/2010 18:17:34 8:00 09/21/2010 00
RRS_RSTOFFLOADSIZE 09/20/2010 18:17:34 8:00 09/21/2010 00
RRS_STORAGE_NUMLARGELOGBLKS 09/20/2010 23:42:49 0:05 09/20/2010 23
RRS_STORAGE_NUMLARGENSBLKS 09/20/2010 23:42:49 0:05 09/20/2010 23
RRS_STORAGE_NUMSERVERREQS 09/20/2010 23:42:49 0:05 09/20/2010 23
RRS_STORAGE_NUMTRANSBLKS 09/20/2010 23:42:49 0:05 09/20/2010 23
RSM_AFG 09/20/2010 18:17:34 ONETIME ***** N/A **
RSM_HVSHARE 09/20/2010 23:37:28 0:15 09/20/2010 23
RSM_MAXCADS 09/20/2010 23:37:28 0:15 09/20/2010 23
RSM_MEMLIMIT 09/20/2010 18:17:34 ONETIME ***** N/A **
RSM_REAL 09/20/2010 18:17:34 ONETIME ***** N/A **
RSM_RSU 09/20/2010 18:17:34 ONETIME ***** N/A **
RTM_LEAVEIT 09/20/2010 18:17:34 ONETIME ***** N/A **
SDSF_CLASS_SDSF_ACTIVE 09/20/2010 18:17:34 ONETIME ***** N/A **
SDSF_ISFPARMS_IN_USE 09/20/2010 18:17:34 ONETIME ***** N/A **
SDUMP_AUTOALLOCATION 09/20/2010 18:17:34 ONETIME ***** N/A **
F1=HELP F2=SPLIT F3=END F4=RETURN F5=IFIND F6=BOOK
F7=UP F8=DOWN F9=SWAP F10=LEFT F11=RIGHT F12=RETRIEVE
    
```

## Additional parameters of the PFA checks

PFA was originally designed to be "black box" with no customizable parameters (to keep it simple), but based on customer request:

- Support for a **MODIFY** command was introduced to provide additional information and control
- More detailed information, parameters, and documentation was made available

## Additional parameters of the PFA checks

Each check can have additional check-specific parameters

- For example to specify a certain threshold to PFA, to reduce "false positives", or define deviation settings

There are two ways to change the parameters for a check:

- Using the **HealthChecker MODIFY** command as discussed earlier. This is the mechanism used for *nearly every* parameter:
  - The parameters for each check are listed in the z/OS Problem Management book
- The **MODIFY PFA,UPDATE** command was introduced specifically for use with the **EXCLUDE\_JOBS** parameter, because the parameter is potentially longer than is supported by Health Checker on its **MODIFY** command. It is intended to drive PFA to refresh its in-storage list from the file you specify on the command.

## Available checks in previous releases - part 1

### Common storage usage (shared resource exhaustion)

- CSA + SQA usage below the line
- ECSA + ESQA usage above the line
- Reports which job is likely to be the cause

### Logrec arrival rate by key (damaged system or address space)

- Provide a list of possible jobs that caused software failures

### Frames and slot usage (shared resource exhaustion)

- Increased usage by persistent (started within one hour after IPL) address spaces
- Reports by address space - these are available when you look at the check output in SDSF CK.

## Available checks in previous releases - part 2

### Console message arrival rate:

- The number of messages per CPU used second for:
  - The total system rate
  - The top individual persistent jobs, that have the highest rate
  - For other persistent jobs as a group
  - For non-persistent jobs as a group

## New PFA enhancements in z/OS V1R12

**A new check was added:**

- The SMF record arrival rate check

**Based on customer feedback**

- A new "Supervised Learning" capability was introduced

**Many performance, modeling, serviceability, and usability improvements in areas such as:**

- Common storage usage check
- Logrec arrival rate check
- Dynamic modeling algorithms
- Log files

## SMF record arrival rate (damaged system)

**This check will track the arrival rate of SMF records in a time range, normalized by CPU usage**

**Collects, models, and compares are done in four different categories:**

- "Chatty" persistent address spaces (those started within 1 hour of IPL)
- Non-chatty persistent address spaces
- Non-persistent address spaces
- Total system rate

**The report will provides a list of jobs that caused the spike in the rate of SMF record creation**

**The STDDEV parameter and the EXCEPTIONMIN parameters can be used to adjust the sensitivity of the comparisons**

## Supervised Learning support

To increase the accuracy of PFA and reduce the number of false positives, PFA now supports both unsupervised learning and supervised learning:

- Unsupervised learning is the machine learning that PFA does automatically.
- Supervised learning allows you to exclude jobs that are known to cause false positives. For example:
  - Exclude test programs that result in many LOGREC records and cause exceptions
  - Exclude address spaces that issue many WTOs, or are inconsistent or spikey in their behavior and cause message arrival rate exceptions
- An EXCLUDED\_JOBS file is used to specify jobs the check should exclude
- This file is located in the check's /config directory. For example,
  - /u/pfauser/PFA\_MESSAGE\_ARRIVAL\_RATE/config/EXCLUDED\_JOBS

## Improved modeling - part 1

- The Common Storage usage check has improved granularity
  - Previously, PFA monitored SQA + CSA (below the line) and ESQA + ECSA (above the line)
  - These are now separated into:
    - SQA
    - CSA
    - ESQA
    - ECSA
- Common Storage usage check and LOGREC arrival rate check have performance improvements
- LOGREC arrival rate check added EXCEPTIONMIN as new configuration parameter, to give the ability to set a threshold



## Improved modeling - part 2

- Checks are enhanced to improve dynamic modeling and comparison algorithms
  - Improve performance by reducing the number of model requests for a stable system (as determined by PFA)
    - Every 720 minutes (12 hours) by default instead of every 6 hours in previous releases
    - PFA dynamically determines when to model more frequently based on system behavior
      - If the system is stable, it updates its prediction once every 12 hours for each check
      - If the system is not stable, the prediction interval will be adjusted to once every 6 hours.
  - The LOGREC arrival rate - allow data to be used across an IPL and not require a 24 hour warm-up period

## Serviceability improvements

Prior to z/OS 1.12, several log files existed in the /data directory, but the same files were used for different types of processing. Starting in z/OS 1.12:

- Each check will have its own subdirectory and log files for each step in the processing
  - CONFIG.LOG, COLLECT.LOG, MODEL.LOG, and RUN.LOG
  - Other log files specific to the check are also created
- If PFA issues an exception for a potential problem, log files and data files needed to investigate the exception are copied to a new directory named "EXC\_timestamp" which is created in the check's directory
  - An additional REPORT.LOG is created which contains the information written to the health checker report
  - The files for the last 30 exceptions are stored. If more than 30 exceptions are issued, the oldest EXC\_ directories are deleted

## PFA and Health Checker modify command

PFA does not have its own Parmlib member, even though each "check" has a number of parameters that can be overridden.

The only way to modify these parameters on a "permanent" basis is to update the HZSPRMxx member with the overrides that you wish to use.

## New command option and messages

### New options for the PFA modify command

- F PFA,UPDATE to request PFA to read a (changed) excluded job file

### New messages for the SMF arrival rate check

- AIRH187E
- AIRH188E
- AIRH191E
- AIRH174E

**STRONGLY** recommend to review your automation policy to ensure that all the messages listed in the z/OS Problem Management book are presented to the operator (at a minimum)

## Installation and migration

The complete installation and migration is described in Chapter 7 of the z/OS Problem Management manual - read through the whole chapter BEFORE you start the install.

- The installation requires some knowledge of the USS
- Note that the migration process has changed with R12
- The manual omits to mention that SMF exits IEFU83/84/85 must be enabled, and the CSA Tracker (in DIAGxx) must be active

The file structure is changed from V1R11, where every check had its own ini file, to V1R12, where only one ini file is required in the etc/PFA directory.

The parameters for each check are described in Chapter 9 of the z/OS Problem Management manual.

## PFA Quirks

- It appears that if you restart the PFA address space, any overrides that you had specified to HealthChecker before PFA was shutdown do not automatically get picked up by PFA when it is restarted.
- You might get different results if you look in CK and do a F PFA,DISPLAY - the information provided in the check in CK only gets updated when the check runs, so if you change a parm, the change will not show in CK immediately.
- If you restart PFA, the effect is the same as an IPL. That is, it will not do a compare until after the next scheduled modeling run.

## Information and documentation

*z/OS Problem Mangement, G325-2564-07. Part 3: Predictive Failure Analysis* is the official home for all PFA documentation

August 2010 issue of *z/OS Hot Topics*, available from:

[http://publibz.boulder.ibm.com/zoslib/pdf/e0z2n1b0\\_7.26.pdf](http://publibz.boulder.ibm.com/zoslib/pdf/e0z2n1b0_7.26.pdf)

PFA overview (based on z/OS V1R11) gives a good explanation of the concepts, see:

[http://publib.boulder.ibm.com/infocenter/ieduasst/stqv1r0/index.jsp?topic=/com.ibm.iea.zos/zos/1.11/Availability/V1R11\\_PFA/player.html](http://publib.boulder.ibm.com/infocenter/ieduasst/stqv1r0/index.jsp?topic=/com.ibm.iea.zos/zos/1.11/Availability/V1R11_PFA/player.html)

There is also an APAR (OA33776) to help determine the size of the PFA database

– On my sysplex it was about 300 tracks per system



## Runtime Diagnostics (RTD) support

To analyze a system in "sick but not dead" situation

Especially for use by the systems programmer

- Intent is to perform a quick (1-2 mins) analysis of your system and point you in the right direction
- Intent is to assist you analyzing critical stress situations
- Also designed to help less skilled sysprogs

## Runtime Diagnostics

RTD does not run all the time - it is intended to be started when you detect that the system is not behaving "normally"

- Could be started by an operator when they suspect a problem. OR
- Could be started by automation when one of the documented "critical" messages is issued

What exactly does it do?

RTD performs the following analysis (typically taking <1 minute), trying to emulate the actions of an experienced system programmer:

## Runtime Diagnostics situation analysis

### Critical message analysis

- Checks the last hour of the OPERLOG for critical messages
- Can perform additional checks, based on the messages and the contents of the messages that it finds

### Looks for ENQ contention by:

- Issuing a D GRS,AN,WAITERS
- Compares this with system address spaces started during IPL
- If contention is found; reports the waiter and the blocker
- RTD documentation refers you to other manuals for additional guidance

## Runtime Diagnostics situation analysis

### Performs CPU analysis

- Checks for address spaces that are using more than 95% of the capacity of a single cpu (reported percentage can be greater than 100% if multiple TCBS are having a high CPU usage)

### Local lock suspension

- Checks for address spaces suspended for more than 50% of the time, waiting for a local lock

### Loop detection

- All tasks, in all address spaces are checked.
- To find a task that is looping, various system indicators are checked for consistent repetitive activity.
- When both high CPU usage and a Loop event are found for the same, it is considered to be looping.

## Runtime Diagnostics situation analysis

**Additional analysis is carried out for a number of XCF messages and events**

- In the area of system partitioning, stalled member detection, failed connectors, and various structure-related events

## Runtime Diagnostics situation analysis

**Results of analysis are sent to MVS console and to a data set  
Message prefix is HZR - recommend that you add this to your  
automation to ensure the messages are routed to the correct  
people**



## Preparation

- The HZR member is shipped in `SYS1.IBM.PROCLIB` - Copy the member to your `SYS1.PROCLIB` if required
  - The membername must remain as HZR
- Always start HZR with `SUB=MSTR`, or update the parmlib `IEFSSNxx` member to define HZR as a subsystem: `SUBSYS SUBNAME(HZR)`
- While `OPERLOG` is not *required*, it IS suggested because it is required for analysis of critical messages
  - Make sure you provide HZR with read access to `SYSPLEX.OPERLOG`
- Recommend to assign HZR to `SYSSTC` service class to ensure it gets enough CPU to analyze loop problems

## Runtime Diagnostics support

RTD is shipped as part of z/OS V1R12.

It is not rolled back to earlier releases.

However, it **CAN** do partial analysis of problems on systems other than the system where it is started.

- Can do partial analysis of systems as far back as z/OS V1R10.

## Usage

### Start Runtime Diagnostics with:

- S HZR
- S HZR,SUB=MSTR if no specification in IEFSSNxx
- S HZR,OPTIONS(SYSNAME=abcd) to attempt analysis of a problem on another system
- A DEBUG option is provided. However its objective is to create one or more dumps for analysis of RTD problems. It does not perform additional checks or present more information (use this option only under direction of IBM support).

### For detailed documentation, report explanation and guidance see:

- G325-2564-07 z/OS Problem mangement

## Two examples of a system without problems

### Running on the target system in the Sysplex

S HZR,SUB=MSTR or S HZR,SUB=MSTR,OPTIONS=(SYSNAME=#@\$3) results in:

```
HZR0201I RUNTIME DIAGNOSTICS SUCCESS. TIME (2010/09/05 - 11:59:56).
NO EVENTS WERE FOUND FOR SYSTEM: #@$3
```

### Running from one system but targeting another system in the Sysplex

S HZR,SUB=MSTR,OPTIONS=(SYSNAME=#@\$A) results in:

```
HZR0200I RUNTIME DIAGNOSTICS RESULT 701
SUMMARY: SUCCESS - NO EVENTS FOUND
REQ: 001 TARGET SYSTEM: #@$A HOME: #@$3 2010/09/05 - 11:53:15
INTERVAL: 60 MINUTES
EVENTS:
FOUND: 00 - PRIORITIES: HIGH=00 MED=00 LOW=00
PROCESSING BYPASSED:
LOOP.....SPECIFIED TARGET SYSTEM IS NOT THE HOME SYSTEM.
HIGHCPU...SPECIFIED TARGET SYSTEM IS NOT THE HOME SYSTEM.
LOCK.....SPECIFIED TARGET SYSTEM IS NOT THE HOME SYSTEM.
```

## An example of a system with problems

### S HZR,SUB=MSTR results in:

HZR0200I RUNTIME DIAGNOSTICS RESULT 842

SUMMARY: SUCCESS

REQ: 001 TARGET SYSTEM: #@\$A HOME: #@\$A 2010/09/07 - 16:56:31

INTERVAL: 60 MINUTES

EVENTS:

FOUND: 02 - PRIORITIES: **HIGH=02** MED=00 LOW=00

TYPES: **LOOP=02**

-----  
EVENT 01: HIGH - LOOP - SYSTEM: #@\$A 2010/09/07 - 16:56:31

ASID: 0135 JOBNAME: CPUPIG TCB: 007E6230

STEPNAME: CPUPIG PROCSTEP: TSO JOBID: STC19998 USERID: STC

JOBSTART: 2010/09/07 - 16:56:11

ERROR: ADDRESS SPACE APPEARS TO BE IN A LOOP.

ACTION: USE YOUR SOFTWARE MONITORS TO INVESTIGATE THE ASID.

-----  
EVENT 02: HIGH - LOOP - SYSTEM: #@\$A 2010/09/07 - 16:56:31

ASID: 0136 JOBNAME: CPUPIG TCB: 007E6230

STEPNAME: CPUPIG PROCSTEP: TSO JOBID: STC19999 USERID: STC

JOBSTART: 2010/09/07 - 16:56:18

JOBSTART: 2010/09/07 - 16:56:18

ERROR: ADDRESS SPACE APPEARS TO BE IN A LOOP.

ACTION: USE YOUR SOFTWARE MONITORS TO INVESTIGATE THE ASID.



©2010 IBM Corporation. All rights reserved.



International Technical Support Organization

ibm.com/redbooks

Miscellaneous enhancements and info



© 2010 IBM Corporation. All rights reserved.

## Disablement of "CF Hint" function

### APAR OA31601 disabled the "CF Hint" function

- The **SITE** keyword in CFRM policy will still be accepted, but XES will not do anything with it.
- Only impacts GDPS customers

## GDPS K System Local Timing Mode

XCF was changed in z/OS 1.11 so that if it lost access to the sysplex time, the GDPS K system would go into local timing mode, rather than spinning, waiting for the signal to be restored

In support of this, APAR OA32236 changes SFM so that a system in local timing mode will not partition other members of the sysplex for 80 minutes

## Recent Redbooks

*z/OS Version 1 Release 12 Implementation (draft)*

*IBM zEnterprise System Technical Guide (draft)*

*System z Parallel Sysplex Best Practices, SG24-7817 (draft)*

*SMF and System Logger - Exploiting The New Paradigm,  
SG24-7919 (under development)*

*Server Time Protocol Recovery Guide, SG24-7380*

*System z on the Go: Accessing z/OS from Smartphones,  
SG24-7836*

## Hot Topics magazine

Hot Topics magazine is written by System z developers

Typically 40-60 pages, with each article being 3-4 pages.

Two issues per year, normally released in the week of SHARE.

Available for download from:

- [http://www.ibm.com/systems/z/os/zos/bkserv/hot\\_topics.html](http://www.ibm.com/systems/z/os/zos/bkserv/hot_topics.html)

**HIGHLY** recommended - excellent for providing an introduction for topics outside your area of expertise.



## Recent and interesting service



## Interesting APARs

- OA26986: NEW FUNCTION - NEW XES FUNCTION IN SUPPORT OF IRLM APAR PK75723
- OA30300 - VSAM/RLS Type 42 rec incorrect
- OA31331 - Address WAIT120 due to XCFAS page fault during HyperSwap
- OA32650 - Increase in number of connectors not picked up when new CFRM CDS switched in
- OA33029 - Support for PPRC, FC, GM commands to devices in subchannel set 1
- OA33101 - Update to dynamic exits support to support data sets on EAV volumes
- OA31648 - VSAM RLS Measurement data for buffer latch contention
- OA33285 - Measure the performance of VSAM record management

## Interesting APARs

- OA33421 - Abend in VTAM during structure rebuild, resulting in IXL040E
- OA33449 - Hung systems in sysplex
- PM09519 - z/OS MF New Function toleration PM00068 - Support 100,000 open data sets in DB2
- PM06632 - IMS CQS structure rebuild started unnecessarily
- PM05887 - Problem with CQS structure when defined as non-recoverable
- PM08700 - CQS response to ENF35 causes contention on CFRM CDS
- Recent Red Alerts.
  - <https://www14.software.ibm.com/webapp/set2/sas/f/redAlerts/home.html>

## Hardware maintenance

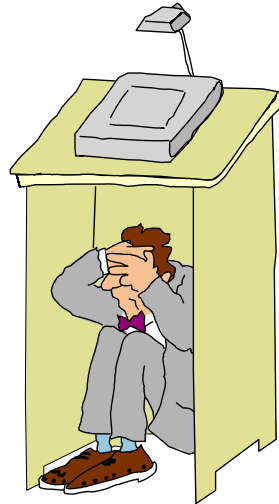
**Driver-76 is now in 'End of MCL Support' mode. No additional MCLs will be released for Driver-76 after July 2010.**

**Driver-79 became available November, 2009 and the majority of z10 machines are currently running with Driver 79 at this time.**

**Product Engineering strongly recommends that you plan to upgrade to Driver-79 during your next microcode maintenance window.**



# Questions?



# Thanks!!



And please remember to hand in your evaluations